

# Vorlesung Automatisierungstechnik

für VerfahrenstechnikerInnen

HS Furtwangen University, Fachbereich Maschinenbau und Verfahrenstechnik

Prof. Dr. Stefan von Weber

Version März 2012 (Skript 1333)

## Inhaltsverzeichnis

## Content

Inhaltsverzeichnis	Content
1. Einleitung	Introduction
2. Automatisierungskonzepte	Concepts of automation
2.1 Gerätesteuerung mit $\mu$ -Controller	Device control with a $\mu$ -controllers
2.2 Laborautomatisierung mit PC	Laboratory automation with a PC
2.3 Anlagenautomatisierung mit SPS	Plant automation with a SPC
2.4 Leitsysteme	Control systems
3. Kommunikation	Communication
3.1 Signaltransport und Verarbeitung	Signals - transport and processing
3.2 Netze	Networks
4. Pneumatik	Pneumatics
4.1 Logik- und Schaltzeichen	Logic and switching symbols
4.2 Pneumatikplan	Pneumatic circuit diagram
4.3 Sensoren	Sensors
4.4 Aktoren	Actuators
5. Vertiefung Regelungstechnik	Deepening of control technique
5.1 Regeln mit der SPS	Control with SPC
5.2 Fuzzy Control	Fuzzy control
5.3 Neuronale Netze	Neuronal networks
5.4 Mehrgrößenregelung	Multivariable control loops
5.5 Stabilität von Regelkreisen, Gütemaße, Optimierung	Stability of control loops Measures of quality, optimization
6. Bildverarbeitung in der AT	6. Image processing in the AT
6.1 Strichcodes	Barcodes
6.2 Bildverarbeitungssysteme	Image processing systems
6.3 Punktoperationen	Point operators
6.4 Fouriertransformation	Fourier transform
6.5 Filter	Filters
6.6 Bildsegmentation	Image segmentation
6.7 Kanten und Linien	Edges and lines
Übungsklausuren, Übungsliste, Dictionary	Written exams, list of exercises, dictionary

## Literatur

## References

- |  |   |
|--|---|
| <b>M. Horn / N. Dourdoumas (2004) : Regelungstechnik, Pearson Studium, München</b> | (1990): Grundlagen der Automatisierungstechnik        |
| Töpfer, Besch  |   |
| <b>Wellenreuther/Zastrow (2001): Automatisieren mit SPS, Vieweg</b>                | (1991): Pneumatische Steuerungstechnik                |
| Haug   |   |
| Dorf/Bishop  | (2005): Modern Control Systems, Pearson Prentice Hall |
| <b>G.Stormann (1990): Messtechnik im Chemiebetrieb, Oldenbourg</b>                 | (1994): Regelungstechnik und Simulation, Vieweg       |
| A.Makarov  |   |
| Zakharian u. a.  | (1998): Neuronale Netze für Ingenieure, Vieweg        |

### 1. Einleitung

Vergleicht man ein Automatisierungssystem mit dem menschlichen Körper, dann sind die Rechner (PC, SPS,  $\mu$ -Controller) das Gehirn, die Signalleitungen die Nervenbahnen, die Sensoren die Augen und Ohren, die Aktoren die Muskeln, die Starkstromleitungen die Adern. Alle Teile sind wichtig und müssen zueinander passen.

Der Aufbau dieser Vorlesung ist auch im Zusammenhang mit der parallel laufenden Vorlesung Regelungstechnik zu sehen. Dinge sollten möglichst nicht doppelt erklärt werden.

### 2. Automatisierungskonzepte

Das **Konzept** hängt von vielen Gesichtspunkten ab, z.B.

- den geforderten Funktionen (Steuern, Regeln, Daten sammeln, ...)
- der Größe der Anlage (Gerät, Anlage, Fabrik, Verbund, ...)
- der Fahrweise (Batch, kontinuierlicher Prozess)
- der geforderten Sicherheit (normal, explosionsgeschützt)

#### 2.1 Gerätesteuerung mit $\mu$ -Controller

Ein  $\mu$ -Controller ist ein 1-Platinenrechner von der Größe eines Handys oder kleiner. Ein  $\mu$ -Controller kann rechnen, Daten speichern, Analogwerte in Zahlen wandeln und zurückwandeln. Man wendet ihn in vielen Geräten an, z.B. in Messgeräten, Analysegeräten, Beschriftungssysteme, Erkennungssysteme. Die  $\mu$ -Controller werden fast ausschließlich in der Programmiersprache C programmiert. Das wichtigste Programmierelement der Steuerung und Regelung ist die zentrale Steuerschleife. Meist nimmt man dafür eine do-while-Schleife der folgenden Form:

Setze Startwerte, wie z.B. Startzeit festhalten oder Stückzähler auf Null setzen

Eröffne Dateien für Protokoll und Daten

```
do {
  Lies Signale von Sensoren, z.B. Lichtschranken, Thermometer, Füllstand, ...
  Berechne mit logischen Funktionen UND, ODER, NEGATION die
  Schaltstellungen für digital angesteuerte Aktoren (EIN oder AUS), wie z. B.
  Füllventile, Ablassventile, Deckelöffner, geschaltete Heizungen,
```

Signallampen usw.

Berechne nach DIN-Algorithmen die Stellgrößen für geregelte Aktoren, z.B.

Regelventile, geregelte Rührermotoren, Dosierpumpen, geregelte Heizungen

Gehe in eine Warteschleife, bis die festgelegte Zeit (Tastzeit) für einen

Schleifendurchlauf um ist, z.B. TA=0,1 Sekunde

**}while** ( kein Stoppsignal und kein Fehler vorliegt)

Mache Aufräumarbeiten, Endstände von Zählern abspeichern, Dateien abschließen

(close), Ventile schließen, Motore abschalten, usw.

Ein Beispiel sei der Autopilot eines Motorbootes. Ein **Sollkurs** ist als Gradzahl vorgegeben ( $0^\circ$ =Norden,  $90^\circ$ =Osten). Der Autopilot hält den **Kurs**, indem er das **Ruder** zwischen  $-20$  und  $+20$  verändert. Die Signalein- und -ausgabe wird hier in Funktionen versteckt. Wegen der Ungenauigkeit bei den GPS-Daten werden die Daten gemittelt (exponentielle Glättung).

```

Ruder = 0; // Startwert der Stellgröße
Eingabe_GPS_Daten(L_alt, B_alt); // L = Länge in Grad, B = Breite in Grad
cout<<"Bitte Sollkurs eintippen:"; // Aufforderung am Display
cin>>sollkurs; // Eingabe von der Tastatur abholen

do { Eingabe_GPS_Daten(L_neu, B_neu); // Eingabe alle Sekunde

    dx = (L_neu - L_alt) * cos(B_alt); // Längengradbewegung des Schiffes
    dy = (B_neu - B_alt); // Breitengradbewegung

    if (dy!=0) {roh_kurs = arctan(dx / dy);} // Rohkurs berechnen
                else {if(dx>=0) {roh_kurs=PI/2;} else [roh_kurs=3*PI/2;]}
    if (dy<0) roh_kurs = roh_kurs + PI; // 1. bis 4. Quadrant
    roh_kurs = roh_kurs * 180.0 / PI; // Rohkurs in Grad umrechnen
    roh_kursdiff = sollkurs - roh_kurs; // Kursabweichung feststellen

    // Nordrichtung macht Ärger, z.B. Sollkurs=1°, Rohkurs=359°
    if (roh_kursdiff < (-300)) roh_kursdiff = roh_kursdiff +360;
    if (roh_kursdiff > (+300)) roh_kursdiff = roh_kursdiff -360;

    kursdiff = 0.99 *kursdiff +0.01 * roh_kursdiff; // Exponentielle Glättung

    Ruder = 1.5 * kursdiff ; // Einfacher P-Regler mit KPR=1,5
    if (Ruder > 20 ) ruder = 20; // obere Beschränkung der Stellgröße
    if (Ruder < (-20 )) ruder = -20; // untere Beschränkung der Stellgröße
    Analogausgabe(kanal_0, Ruder); // Ansteuerung Rudermaschine
    delay(1); // Warte 1 Sekunde
    autopilot_aktiv = Eingabe_Schalterstellung ( Schalter_1);
} while (autopilot_aktiv);

cout<<"Autopilot nicht aktiv";

```

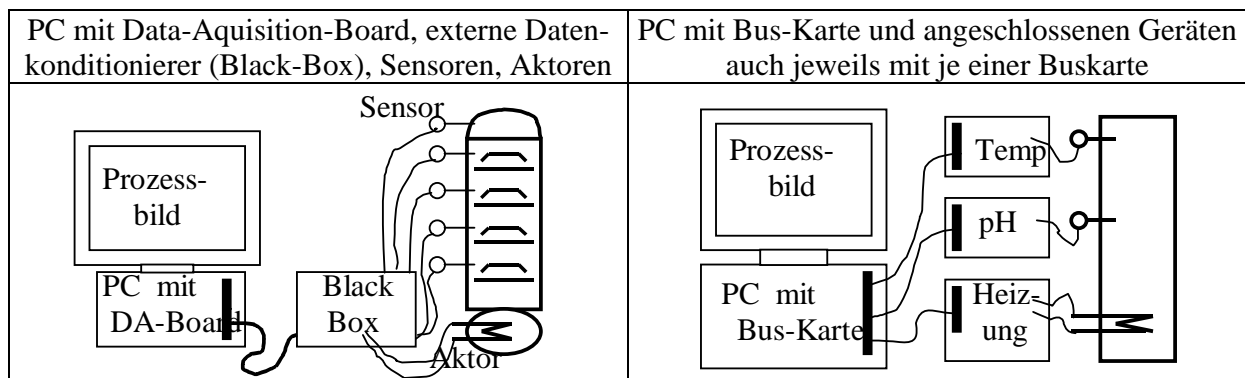
Einige Dinge, die es beim  $\mu$ -Controller zu beachten gilt, sind z.B.:

- Man braucht ein Netzgerät mit mehreren Spannungspegeln (5 V, 12 V)
- Man braucht eine Entwicklungsumgebung (einen PC mit C-Compiler, Bibliotheken)
- Man braucht ein Gerät zum Übertragen des Programms in den  $\mu$ -Controller (Brenner)
- Der  $\mu$ -Controller lässt sich nach einem Löschvorgang (erase) beliebig oft neu brennen
- Die Spannungen und Stromstärken der Ausgangssignale sind schwach (5 Volt, 1 mA)
- Der  $\mu$ -Controller ist empfindlich für Überspannungen
- Der  $\mu$ -Controller rechnet deutlich langsamer als ein PC

**Siemens Logo!** Als Beispiel einer Zwischenstufe zwischen  $\mu$ -Controller und SPS. Dieses kleine Bauteil ersetzt Schütze und Relais im Schaltschrank (z.B. für Verriegelung von Schutzeinrichtungen), kann kleinere 1-phasige Lasten schalten, spart dadurch Verdrahtung im Schaltschrank. Hat Analogausgänge und sogar PI-Regler, z.B. für Heizung, Druck, Füllstand.

## 2.2 Laborautomatisierung mit PC

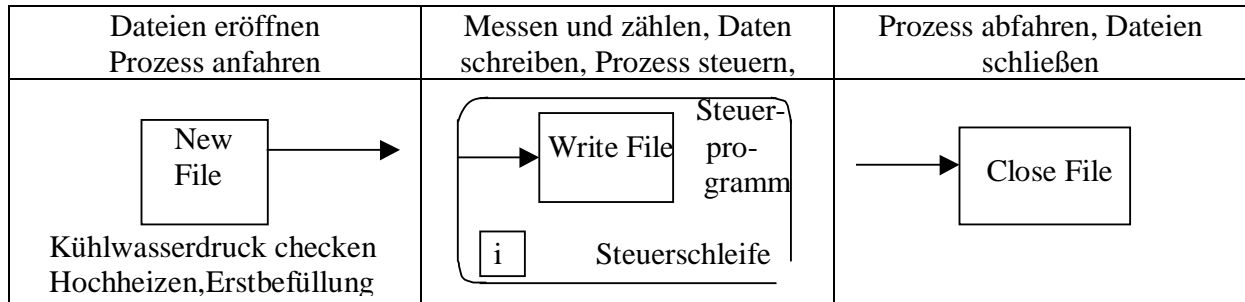
Für eine Laborautomatisierung werden fast ausschließlich aufgerüstete PCs verwendet. Um Daten von und zu Geräten oder Versuchsaufbauten zu transportieren, ist zusätzliche Hardware nötig. Übliche Varianten sind z.B.:



Es gibt zahlreiche Steuerprogramme, die auf den PCs laufen. Sie kommunizieren über das Data-Aquisition-Board oder über die Bus-Karte mit dem Prozess. Immer dazwischen geschaltet sind Signalkonditionierer, die verschiedene Funktionen haben, z.B.

- Verstärkung schwacher Signale bei Eingängen, Relais oder Verstärker bei Ausgängen
- Glättung und Filterung der Signale gegen Rauschen oder Spannungsspitzen
- Schutz vor Überspannungen
- Netzteile für die Stromversorgung

Eine typische Anwendung besteht aus einem Prozessbild auf dem Bildschirm des PC (der Bedienoberfläche) mit eingeblendeten Zahlenangaben, Graphiken von wichtigen Messwerten, Signallampen, Schaltern, Schieberegler usw., und einem Steuerprogramm im Hintergrund, das sowohl mit dem Prozessbild kommuniziert, als auch mit dem Prozess (Signale von Sensoren liest und Aktoren ansteuert). Viele dieser Programme werden graphisch programmiert. Die Grundstruktur eines solchen Programms ist jedoch wieder eine Dreiteilung:



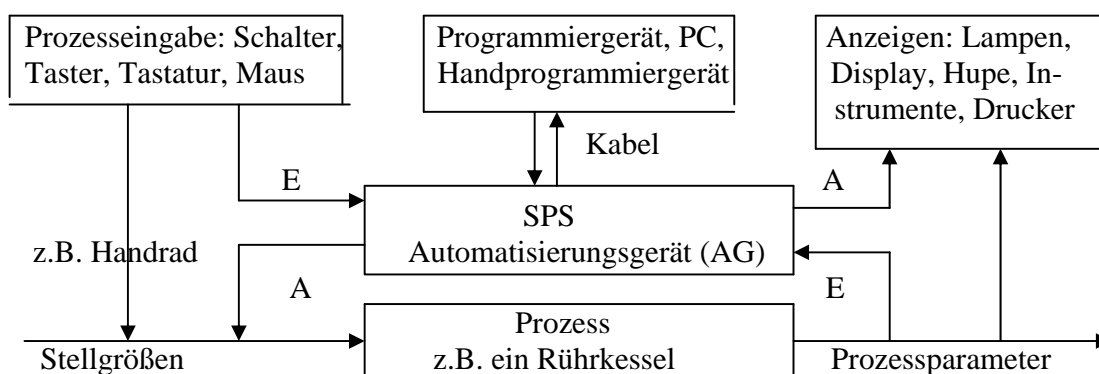
**Industrie-PC** sind heute voll gekapselt, haben keine beweglichen Teile (Lüfter, Laufwerke) und max. 10 Watt Leistungsaufnahme (z.B. mit Prozessor Intel <sup>®</sup>Atom™ Z530 mit 1,1 GHz Taktfrequenz, 2,5 Watt Leistung) und damit sehr robust. Betriebssysteme sind Windows, CE, XP.

### 2.3 Anlagenautomatisierung mit SPS

Die SPS ist das Arbeitspferd der Automatisierer. Sie besteht aus einem robusten Rechner und sehr leistungsfähigen eingebauten Signalkonditionierern. Der externe Spannungspegel für digitale Signale (0-1-Signale) ist 24 Volt. Die Stromstärke kann bei der Ausgabe bis zu 1 Ampere betragen. Analoge Daten werden ebenfalls verarbeitet im Spannungsbereich -10 bis +10 Volt (bzw. bei Stromsignalen 4 bis 20 mA Industriennorm). Die SPS ist entweder in einem Schaltkasten vor Ort direkt neben der Anlage untergebracht, oder aber in einem Schaltschrank in einer separaten Schaltwarte. Die Prozesssteuerung erfolgt zumeist über ein Touchpad.

Im Skript *Regelungstechnik* erfahren Sie mehr über den Aufbau einer SPS, ebenso über den Aufbau eines Steuerprogramms. Die unten stehende Graphik stammt aus dem Skript *Regelungstechnik*. Im **Kapitel 5** des vorliegenden Skriptes können Sie dann nachlesen, wie eine SPS mit Analogsignalen umgeht und man einen Regelkreis aufbaut.

#### Zusammenspiel Prozess mit SPS



### 2.4 Leitsysteme

Leitsysteme sind eine hierarchische Kombination aus einem Leitreechner (meist ein PC), untergeordneten Steuerrechnern (meist SPSen) und oder Industrieregler. Das Wesentliche am Leitsys-

tem ist, dass auf dem Bildschirm oder den Bildschirmen des Leitrechners alle wichtigen Informationen zusammen laufen. Der Prozess, z.B. eine Brauerei, ist auf verschiedenen Prozessbildern dargestellt, z.B. das Sudhaus, der Gärkeller, der Lagerkeller, die Abfüllanlagen, das Auslieferungslager, das Leergutlager, das Qualitätslabor. Per Mausklick kann der Betriebsleiter sich nicht nur alle Prozessstände ansehen (Füllstände, Temperaturen, Lagerbestände, Zahl der aktiven Gärtanks), sondern der Braumeister kann vom Leitrechner aus Batchprozesse starten, wie das Befüllen der Sudpfanne mit Wasser und Würze und das anschließende Kochen und Klären.

### 3. Kommunikation

Ein immer mehr wachsender Anteil an Automatisierungskosten fällt in das Gebiet der Kommunikation. Kommunikation beginnt bereits im Gerät, z.B. einem Messumformer direkt an einem Thermometer, wo die einzelnen elektronischen Bauelemente, wie Prozessor, Speicher, Display, durch interne Busverbindungen (Flachkabel mit Bustreibern an jedem Ende) verbunden werden.

Die nächste Stufe der Kommunikation ist die Anbindung von Sensoren und Aktoren, z.B. des Messumformers eines Thermometers oder des Stellungsreglers eines Ventils, über sogenannte Feldbusse an das nächstgelegene Steuergerät, zumeist eine SPS. Bis hierhin spricht man von der **Feldebene** (Anlagennaher Datenaustausch).

Eine weitere Stufe ist die **Kopplung SPS-Leitrechner**. Hier werden hauptsächlich Zahlen und Text ausgetauscht. Die Busse sind für die Übertragung großer Datenmengen ausgelegt. Die SPS meldet Zählerstände, Temperaturen, Ventilstellungen usw. an den Leitrechner. Dieser gibt Rezepturnummern, neue Sollwerte oder Startbefehle für Teilprozesse an die SPS. Diese Ebene wird oft als **MES-Ebene** (Manufacturing Execution System) bezeichnet. Von hier aus werden aber auch Produktionszahlen bzw. Ressourcenanforderungen an die ERP-Ebene gesandt und Aufträge entgegengenommen. Die **ERP-Ebene** (Enterprise Resource Planning) ist in der Geschäftsleitung angesiedelt. Dort werden Maschinenauslastung, Auftragsdurchlauf, Transport- und Lagerauslastung geplant.

#### 3.1 Signale - Transport und Verarbeitung

**Signalformen:** Analoge Signale für sich kontinuierlich verändernde Größen, wie Temperatur, Druck, Drehzahl, pH, Trübung, Füllstand, Position, Konzentration, Leitfähigkeit usw.

Strom	meist 4-20 mA Industriennorm, geeignet für Entfernungen bis 500 m, Spannung schwankend 5-10 V
Spannung	meist -10 bis +10 V Industriennorm, geeignet für Entfernungen bis 10 m (Abschirmung ist wichtig), Stromstärke <1 mA
Druck	0,2 bis 1,0 bar Überdruck Industriennorm, nur noch in alten pneumatischen Anlagen zu finden

Digitale Einzelsignale (Binäre Signale) für mechanische, elektrische oder elektronische Elemente, die nur zwei Werte liefern bzw. annehmen können, wie z.B. Schalter, Taster, Lichtschranken, Stellung eines Absperrventils, Signallampen, Pegelmelder usw.

Strom	0 oder 20 mA Industriennorm
Spannung	0 oder 24 V Industriennorm

Digitale Signale für beliebige Größen und Nachrichten. In Form von Zahlen und Texten, die binär verschlüsselt übertragen werden, können alle oben genannten Informationen übertragen werden. Zusätzlich können in Textform noch Befehle an sogenannte intelligente Messumformer oder Industrieregler übertragen werden, um z.B. Alarmpegel oder Sollwerte zu verändern, oder ein intelligenter Sensor liefert eine Fehlermeldung im Klartext, z.B. "Hilfsspannung U4 ausgefallen".

**Wandler:** Analoge Signale werden in den Messumformern,  $\mu$ -Controllern, SPSen oder im Industrieregler bei der Eingabe in Zahlen umgewandelt bzw. bei der Ausgabe aus einer Zahl rückgewandelt. Wir unterscheiden zwei grundsätzlich verschiedene Wandlerarten:

1. Das Wägeverfahren speichert mit einem elektronischen Halteglied den Augenblickswert  $U_x$  eines Strom- oder Spannungssignals als Aufladung eines kleinen Kondensators. Dann beginnt eine elektronische Schaltung nacheinander Vergleichsspannungen zu erzeugen, z.B.  $U_7=5$  V. Ist die Spannung  $U_x$  im Kondensator größer oder gleich  $U_7$ , wird Bit 7 der zu erzeugenden Binärzahl 1 gesetzt, sonst 0. Im nächsten Schritt wird in Abhängigkeit von der Bitbelegung die nächste Vergleichsspannung erzeugt,  $U_6=2,5$  oder  $U_6=7,5$  V. Jetzt wird wieder mit  $U_x$  verglichen und das Bit 6 gesetzt usw. bis alle 8 Bits gesetzt sind. Standard ist 8 oder 12 Bit, selten mehr bei den Wägeverfahren. Wägeverfahren arbeiten sehr schnell. Man erreicht leicht 10.000 Wandlungen pro Sekunde. Der Nachteil ist, dass Rauschen oder Spannungsspitzen die Messung beeinträchtigen, da Augenblickswerte gewandelt werden.
2. Das integrierende Verfahren lädt einen etwas größeren Kondensator mit der Messspannung  $U_x$  auf. Welligkeiten, Rauschen, Spannungsspitzen werden integriert und damit gemittelt. Über eine kalibrierte Gegenspannung wird der Kondensator über einen definierten Entladewiderstand entladen. Die Zeit bis zum Nulldurchgang der Spannung wird mit einem Schwingquarz gemessen. Die Zeit ist proportional zum Messwert. Die integrierenden Verfahren sind genauer (bis 20 Bit), aber dafür auch deutlich langsamer (10 – 100 Messungen pro Sekunde).

**Zahlen:** Treten als Konstanten, Merkmalswerte oder Signalwerte in den Rechnern auf. Intern werden Binärzahlen verwendet, extern werden Zeichenketten verwendet. Binäre Zahlen haben die Basis 2 im Gegensatz zum Dezimalsystem mit Basis 10.

Beispiel:  $137 = 128 + 8 + 1 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$   
 oder binär                    1        0        0        0        1        0        0        1

Bei negativen Zahlen benutzt man zumeist das 2-er-Komplement. Um die Zahl  $-137$  darzustellen, beginnt man mit der Binärdarstellung der  $+137$ . Wir benutzen ein 16-Bit-Wort (2 Byte)

Binärdarstellung der positiven Zahl	0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1
Negation jedes einzelnen Bits	1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0
Addition einer 1 nach den Regeln des binären Rechnens	1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1

Gleitkommazahlen, d.h. solche mit Dezimalstellen und/oder einem Exponenten, werden ebenfalls intern binär verschlüsselt. Hier werden jedoch die eigentliche Zahl (Mantisse) und der Exponent getrennt als Binärzahlen behandelt.

Für die Ausgabe auf ein Display müssen die Zahlen von der Binärdarstellung wieder in die Dezimalform gebracht werden. Hierbei wird jede Ziffer und jedes Zeichen der Dezimalzahl einzeln als druckbares Zeichen mit einem Zeichencode verschlüsselt (Z.B. der ASCII-Code (American Standard Code for Information Interchange)). Es entstehen Zahlen-Zeichenketten wie:

3      -77    3.14    27E+03 (lies  $27 \cdot 10^3$ )      -3.14e-22 (lies  $-3.14 \cdot 10^{-22}$ )

**Signaltransport:** Die Transportmöglichkeiten für Signale unterscheiden sich u.a. im Preis, Übertragungsrate, Reichweite und Anfälligkeit gegen Fehler

Transportart	Vorteile	Nachteile
2-Draht-Stromleiter verdreht	billig, bis 500 m, wenig Fehler	niedrige Rate
abgeschirmtes Spannungskabel	billig, bis 10 m, öfters Fehler	niedrige Rate
Hochfrequenz-Kabel (HF)	hohe Rate, bis 100 m, wenig Fehler	teuer
Lichtleiter	hohe Rate, bis 1000 m, kaum Fehler	teuer
Funkverbindung	hohe Rate, bis 10 km, wenig Fehler	teuer

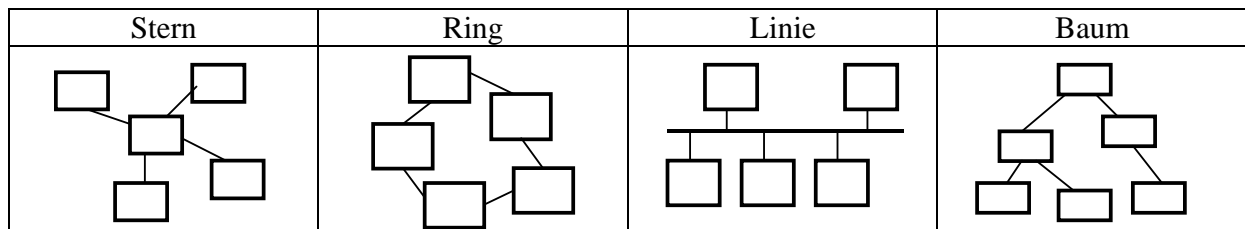
### 3.2 Netze

Netze dienen den Verbindungen Rechner - Sensor, Rechner - Aktor, und Rechner - Rechner.

Ein **LAN** (Local Area Network) darf rechtlich nur auf einem Grundstück arbeiten.

Ein **WAN** (Wide Area Network) darf weltweit agieren unter Beachtung der Posthoheit der betroffenen Länder.

**Netztopologien** sind Stern, Ring, Linie und Baum. Es gibt aber auch Mischformen.



**Übertragungsverfahren** sagen, wie die Signale im Leiter übertragen werden (3 Beispiele):

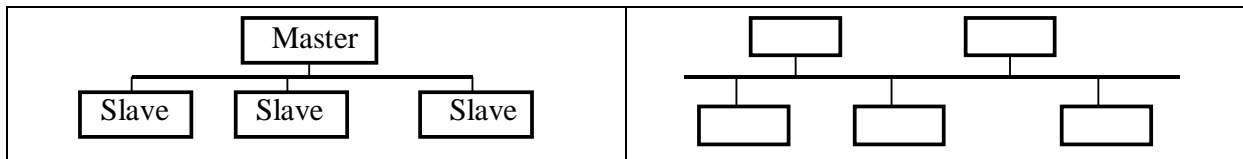
Basisband: 0-1-Signale werden als Spannungspegel übermittelt, z.B. 0 V - 5 V

Trägerband: 0-1-Signale werden als 2 unterschiedliche Frequenzen  $\omega_0$  und  $\omega_1$  übertragen

Breitband: 0-1-Signale werden als 2 Frequenzen  $\omega_0^{(i)}$ ,  $\omega_1^{(i)}$  aus Frequenzband  $i=1, \dots, n$  übertragen, d.h. die Leitung kann mehrere Kanäle gleichzeitig übertragen.

Zugriffsverfahren: Eine Leitung, an der viele Geräte hängen, bedarf einer exakten Regelung, wann welches Gerät senden darf.

Zentrale Bussteuerung: Die Zugriffsberechtigung wird zentral durch den <i>Master</i> vergeben. Die <i>Slaves</i> senden nur auf Befehl.	Dezentrale Bussteuerung: Alle Partner sind gleichberechtigt. Der Zugriff auf die Leitung muss nach einem Vergabeverfahren erfolgen.
---	---



### Zugriffsverfahren (Vergabe der Zugriffsrechte auf die Leitung) (Beispiele):

1. Zentrale Steuerung: Eine Umlaufliste und eine Alarmliste im Masterrechner legen die Reihenfolge der Zugriffsberechtigungen fest. Diese werden in festen Zeittakten vergeben. Ein Slave darf aber in der Liste mehrfach auftreten, was seinen Zugriff verbessert
2. Demand Priority (Basiert auf der DPAM=Demand Priority Access Methode). Alle Geräte sind am Master angeschlossen. Will ein Slave senden, muss es erst eine Anfrage an den Master senden. Der vergibt daraufhin das Zugriffsrecht. Treten mehrere Verbindungsaufträge gleichzeitig auf, werden sie nach Reihenfolge und Priorität des Eingangs vergeben.
3. CSMA/CD (Carrier Sense Multiple Access with Collision Detection): Jeder darf senden, wenn die Leitung frei ist. Bei gleichzeitigem Start zweier Geräte wird die Kollision entdeckt. Jedes Gerät wartet dann eine Zufallszeitspanne lang auf seinen Neustart.
4. CSMA/CA : Baut auf dem CSMA/CD auf, versucht aber Kollisionen zu vermeiden. Sende-willige Geräte schicken Zeichen ins Netz, die von den anderen Geräten beachtet werden. Auch hier spielt die Priorität eine Rolle, z.B. Gerät 001 kommt vor 002.
5. Token Passing (Newhall-Verfahren): Ein Bitmuster (Token genannt) wird von Rechner zu Rechner weitergereicht. Nur der darf senden, der gerade das Token hat. Die zu sendende Information und die Adressen des Zielrechners und des Absenders (d.h. ihre Rechnernummern) werden an das Token angehängen. Der Empfänger setzt in die Nachricht ein Quittungsbit und reicht das Token mit seinem Anhang weiter. Der Absender löscht seine quitierte Nachricht im Tokenanhang, wenn sie wieder bei ihm vorbei kommt.

Die zentrale Steuerung wird bei hoher Netzauslastung empfohlen.

CSMA/CD wird bei niedriger Netzauslastung empfohlen.

Token Passing wird bei mittlerer Netzauslastung empfohlen

### Begriffe aus der Netzwelt:

Repeater	sind in die Leitung eingebaute Signalverstärker bei langen Leitungen
Bridges	sind Verbindungsknoten (Rechner) zwischen unterschiedlichen Zugriffsverfahren, z.B. Token-Passing und Zentrale Steuerung
Gateways	sind Verbindungsknoten (Rechner) zwischen unterschiedlichen Protokollen (z.B. unterschiedliche Spannungspegel oder Signalfolgen beim Handshake)
Router	arbeiten wie Bridges, suchen aber über Adresslisten zusätzlich den Weg zum Zielrechner in einem vermaschten Netz, auch in WAN-Netzen (z.B. WWW)
Protokolle	sind Signalfolgen in einem Dialog zwischen Rechnern, die zur Aufnahme, Durchführung und Beendigung einer Datenübertragung notwendig sind.
Busse	sind Datenleitungen mit Bus-Controllern an jedem Ende
Feldbusse	sind serielle Busse zur Ankopplung von Sensoren und Aktoren an übergeordnete Rechner (z.B. an SPSen oder Leitrechner). Als Übertragungsmedium dient oft nur eine verdrehte elektrische Doppelleitung.

Master	sind aktive Buskomponenten, die auf den Bus zugreifen.
Slaves	sind passive Buskomponenten, die vom Master gesteuert werden.
Profibus	ist ein langsamer Kommunikationsbus. Es werden Texte, Befehle, Zahlen übermittelt. Der Profibus erfordert "intelligente" Slaves, d.h. solche mit eigener Rechenleistung. Es sind viele Stationen mit mehreren Mastern möglich.
Interbus-S	ist ein schnellerer zyklischer Abtastbus mit einem Master und vielen Slaves
CAN-Bus	ist asynchron, seriell, besonders im Automobil vertreten, benutzt CSMA/CA, Kupfer- oder Glasfaser, Multimaster-Architektur mit Übertragungsraten von ca. 1 MBit/s bei Entfernungen bis 500 m, Linientopologie bevorzugt.
ASI-Bus	(Actuator-Sensor Interface), ein Bus für die unterste Automatisierungsebene, Twisted-Pair-Kabel, bis zu 31 Slaves. Es gibt Master-Einschubkarten für SPSen und PCs. Diese fragen die Slaves zyklisch ab. Sternarchitektur bevorzugt.
Ethernet	Kabelgebundenes LAN-Netzwerk. Datenaustausch in Form von Datenrahmen. Festlegung von Kabeltypen, Steckern, Paketformate, Protokollen in Standards.
IGB	Integrated Bus ermöglicht zusätzlich zum Datenaustausch noch Internetzugang und Telefonverbindung für Mails und Nachrichten.

**Wide Area Automation Controller** bieten über verschiedene Kanäle Möglichkeiten der vernetzten Automation. Auf der Feldebene stehen die RTUs (Remote Terminal Unit) mit digitalen und analogen Eingängen, Relaisausgängen für Steuerung vor Ort. Die zentrale Steuereinheit (PC) enthält Ethernet-LAN, ISDN, GSM (Global System for Mobile Communications), USB, RS485 bzw. RS422, Web-Server (Internetzugang), FTP-Server (File Transfer Protocoll), File System (Datenbank)

## 4. Pneumatik

Pneumatik ist die Verwendung von Druckluft als Signalmedium und für Energietransport. Die Anfänge der Pneumatik reichen 150 Jahre zurück (Rohrpost 1853, pneumatische Bohrmaschinen für Tunnelbau und Stahlbau um 1870, Niethämmer um 1900). Die Blütezeit der pneumatischen Steuerung und Regelung war zwischen 1900 und 1980, zuletzt hauptsächlich im Explosionsschutz von Anlagen. Heute werden fast keine pneumatischen Steuerungen mehr gebaut, wohl aber pneumatische Antriebe und Sensoren. Bei den Antrieben ist es die robuste und billige Bauart, bei den Sensoren der Selbstreinigungseffekt und die Unempfindlichkeit gegen elektromagnetische Störfelder, die auch heute noch eine Verwendung sinnvoll machen.

Hauptanwendungsgebiete der Pneumatik sind heute:

- Pneumatische Stellantriebe für Ventile und Klappen, Spannvorrichtungen, Auswerfer, Vorschubeinrichtungen z.B. an Bohrmaschinen
- Düsenspritzgeräte wie Sandstrahlen, Farbsprühen, Späne wegblasen
- Luftdruckhammer
- Pneumatische Regler werden im Fahrzeugbau z.T. noch eingesetzt z.B. zur Drehzahlregelung bei Dieselmotoren und als Stellungsregler in pneumatischen Ventilantrieben.
- Pneumatische Endtaster, Luftschranken, Ultraschallschranken

Für die Verfahrenstechnik und Biotechnologie ist die elektropneumatische Steuerung wichtig. Eine SPS oder ein Rechner erzeugt die elektrischen Steuersignale. Diese schalten elektropneumatische Ventile, und diese wieder die rein pneumatischen Aktoren

Die Vorteile der Pneumatik sind:

- Die a-priori gegebene Überlastsicherung bei Arbeitszylindern, denn nur die Kraft  $F = A \cdot P$  (Fläche mal Druck) kann maximal wirken. Getriebe z.B. sind da gefährlicher.
- Geringe Wartung durch neue verschleißarme Werkstoffkombinationen (Metall - Kunststoff)
- Verfügbarkeit von standardisierten Bauelementen von vielen Herstellern, die auch von Laien ausgetauscht werden dürfen im Gegensatz zur Starkstromelektrotechnik
- Hohe Bewegungsbeschleunigung bei Arbeitszylindern
- Leichte Kraftregulierung durch Drosselventile in den Zu- oder Abluftleitungen

Nachteile der Pneumatik sind:

- Die hohen Luftkosten (Verdichtung, Reinigung, Transport durch Rohrleitungen)
- Der Geräuschpegel beim Luftauslass

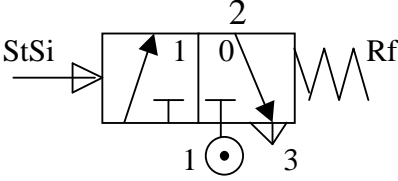
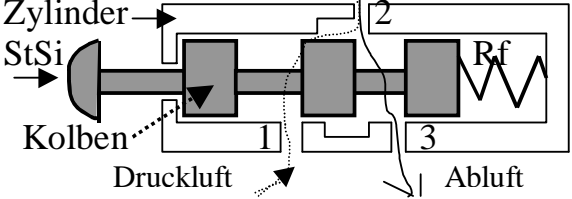
Die Luft muss auf **6 bis 8 bar** verdichtet werden und von Rost, Zunder, Staub, Wasser und Öl gereinigt werden. Man benutzt einen Zyklon und danach Filter mit Porendurchmesser 5-50  $\mu\text{m}$ . Der Druckabfall im Filter liegt bei 0,2 bis 0,3 bar. In alten Anlagen werden noch **Öler** eingesetzt, die die Luft mit einem Ölnebel versetzen. Da Öl giftig ist, arbeiten neuere Anlagen ölfrei.

**Leitungen** sind entweder starr aus Metall oder flexibel aus Kunststoff. Starre Verbindungen sind verschraubt (Fittings), geflanscht, gelötet, geschweißt oder geklebt. Flexible Verbindungen haben fast immer lösbare Schnellkupplungen (Plug and go).

**Ventile** sind das wichtigste Steuerelement der Pneumatik:

- Als Regelventil für den Arbeitsdruck in den Druckluftleitungen der Pneumatik
- Als Sperr- oder Entlüftungsventil für einen Schnellstopp der Anlage
- Als Steuerventil, das je nach Schaltstellung unterschiedliche Luftwege durchschaltet
- Als Leistungsventil, das für große Aktoren große Luftmengen steuert

Beispiel eines Steuerventils: Monostabiler 3/2-Wegekolbenschieber. Der Name sagt, dass das Ventil 3 Luftanschlüsse (1, 2, 3) hat und zwei Schaltstellungen (0 und 1) hat. In der Schaltstellung 0 ist die Entlüftung (3) an die Arbeitsleitung (2) angeschlossen, in Schaltstellung 1 hingegen die Druckluft (1). Die Schaltzeit bei elektromagnetischem Antrieb ist 8–25 ms. Das Spiel des Kolbens im Zylinder beträgt 0,002 – 0,004 mm.

<p><b>Symbol:</b> Steuersignal StSi, Druckluft 1, Arbeitsleitung 2, Abluft 3, Rückstellfeder Rf</p>	<p><b>Mögliche Konstruktion:</b> Im Nullzustand ist Arbeitsleitung 2 mit Abluft 3 verbunden. Bei gedrücktem Knopf ist Druckluft 1 mit der Arbeitsleitung 2 verbunden.</p>
	

### Betätigungsarten des Ventils (Steuertrieb einzeln oder kombiniert)

Vorsteuer- oder Pilotventile ermöglichen es mit kleinen Leistungen größere Arbeitsventile zu schalten. Tasterrollen hingegen fahren auf einer Schablone entlang und steuern so das Ventil.

Durch Luftdruck vorgesteuert	Durch Luftdruck direkt	Elektromagnet stoßend	Elektromagnet stoßend + ziehend	Rückstellen mit Federkraft

Muskelkraft	Schlagknopf	Tasterrolle	Motor

### 4.1 Logik- und Schaltzeichen (Pneumatische Schaltzeichen nach DIN-ISO 1219)

Ruhestellung hat Nummer 0 Fehlt definierte Ruhestellung, dann definiert man die Ausgangsstellung Pro Schaltstellung zeichnet man einen Kasten Die Anschlüsse werden am Kasten 0 gezeichnet	
---	--

#### Verschiedene Steuerventile:

2 / 2	3 / 2	4 / 2	5 / 2

#### Bezeichnung der Leitungen

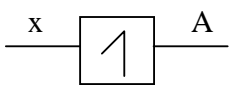
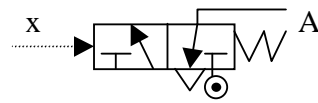
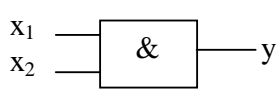
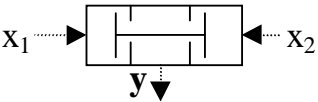
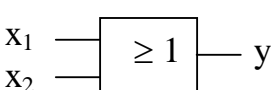
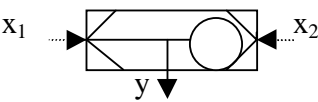
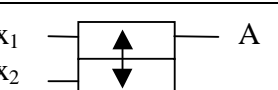
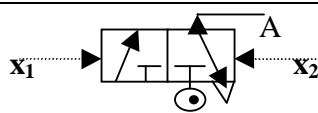
Art der Leitung	Symbol	Nummer ISO 5599 (alte Bez.)
Druckluft		1 ( P )
Arbeitsleitungen		2, 4 ( A, B )
Entlüftungen		3, 5 ( R, S )
Steuerleitungen		12, 14 ( z, y )

Steuersignal 12 (sprich eins-zwei) verbindet Anschluss 1 mit Anschluss 2

Steuersignal 14 (sprich eins-vier) verbindet Anschluss 1 mit Anschluss 4

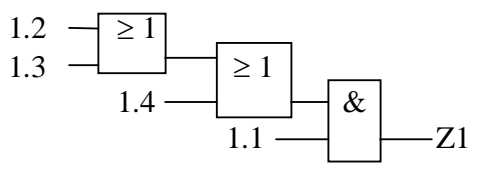
## Pneumatische Verknüpfungen (Auswahl)

Wir unterscheiden logische Schaltzeichen nach DIN 40900 und pneumatische Schaltzeichen nach DIN-ISO 1219. Steuerventile für logische Aufgaben werden heute nur noch dort eingesetzt, wo die Elektropneumatik teurer wäre und keine Zeitvorteile bringt. Bei der Konjunktion (logisches UND) zeigt das pneumatische Schaltzeichen eine bewegliche Sperre aus 2 Scheiben, mit einem Abstandshalter. Bei der Disjunktion (logisches ODER) rollt im pneumatischen Schaltzeichen eine Absperrkugel in einen der Trichter, oder flattert in der Mitte. Identität und Haftspeicher sind 3/2-Steuerventile. Signale  $x$ ,  $x_1$ ,  $x_2$ ,  $y$  sind Steuersignale (passiv),  $A$  ein Arbeitssignal (aktiv).

Funktion	DIN 40900	DIN-ISO 1219
$x = A$ (Identität) Steuersignal $x$ als Leistungssignal $A$ durchgeschaltet		
$x_1 \wedge x_2 = y$ (Konjunktion oder logisches UND) erzeugt ein Steuersignal (schwach)		
$x_1 \vee x_2 = y$ (Disjunktion oder logisches ODER) erzeugt ein Steuersignal (schwach)		
Haftspeicher (Flipflop) schaltet ein starkes Arbeitssignal $A$ ( $x_1$ =Set- und $x_2$ =Resetsignal)		

## 4.2 Pneumatikplan

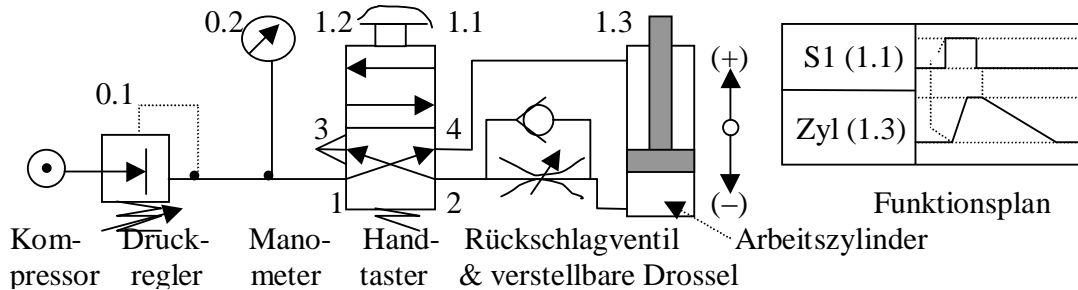
### Logische Schaltungen

Zylinder Z1 soll nur ausgefahren werden, wenn Signal 1.1 <u>und</u> mindestens eines der Signale 1.2 und/oder 1.3 und/oder 1.4 anliegt.	
---	--

Der Pneumatikplan enthält die pneumatischen Schaltzeichen (siehe oben), zusätzlich auch Elemente der Luftbereitstellung, -kontrolle und -freigabe (Kompressor, Druckregler, Manometer, Freigabeventil). In der Nähe von Arbeitszylindern werden Funktionsdiagramme angeordnet, die die Dynamik des Ein- und Ausfahrens des Zylinders graphisch darstellen (Steuersignal über der Zeit aufgetragen, Weg über der Zeit aufgetragen).

Beispiel: Bei Betätigung des Handtasters 1.1 am Steuerventil 1.2 fährt Zylinder 1.3 im Eilgang in die Plus-Lage (+). Nach dem Loslassen des Tasters fährt der Zylinder langsam zurück in die Minus-Lage (-). Das langsame Zurückfahren wird durch eine verstellbare Drossel in der Leitung 2 reguliert. Das Rückschlagventil verhindert das schnelle Abströmen der Abluft. Der Funktionsplan

zeigt diese Dynamik. (Oben das Handtastersignal, unten der Weg-Zeit-Plan. Die auslösenden Signalfanken sind mit dem Weg-Zeit-Diagramm durch feine gestrichelte Pfeile verbunden.)



### Übliche Steuerungssysteme in der Pneumatik

- Manuelle Steuerung (Hand- und Fußbetätigung von Ventilen)
- Zeitabhängige Folgesteuerung (berücksichtigt auch Wartezeiten, z.B. startet der Endschalter eines Zylinders nach einer Wartezeit von 1 Sekunde den nächsten Vorgang)
- Programmschaltwerksteuerung (Nockenwelle mit konstanter Drehung betätigt Ventile)
- Schrittschaltwerksteuerung (Nockenwelle dreht erst auf ein auslösendes Signal weiter)
- Wegabhängige Folgesteuerung (Tasterrollen laufen auf Schablonen und steuern Ventile)
- Taktkettensteuerung (von Zustand zu Zustand, wird heute kaum noch pneumatisch realisiert)
- Elektropneumatische Steuerung (Eine SPS steuert den Einsatz der pneumatischen Aktoren)

### 4.3 Pneumatische Sensoren (Auswahl)

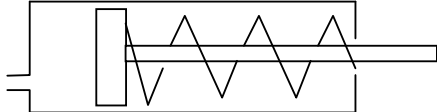
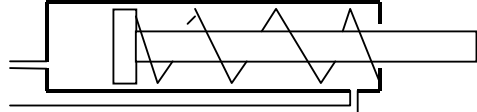
Pneumatische Sensoren haben einen Selbstreinigungseffekt bei Holzspänen oder Staub. Sie sind unempfindlich gegen Hitze, elektromagnetische Felder oder Vibrationen.

<p><b>Luftschranke</b> bis zu einem Abstand <math>A=50</math> mm einsetzbar um z.B. das Beladen einer Hobelmaschine mit einem Brett zu kontrollieren. Der Luftstrom aus der Strahldüse S staut sich im Trichter des Empfängers E. Das auswertbare Drucksignal <math>x</math> wird durch einen Turbulenzverstärker (TV) weiter verstärkt</p>	
<p><b>Näherungsgeber:</b> Bei großer Entfernung des Körpers K zur Düse ist das Signal <math>x</math> ein Unterdruck. Die aus der Düse strömende Luft erzeugt ihn. Bei Annäherung auf 6-8 mm staut sich die Luft. Signal <math>x</math> schlägt um in Überdruck. Die Messgenauigkeit ist 0,1 mm. Ein pneumatischer Membranverstärker erzeugt dann aus <math>x</math> ein verwertbares 0-1-Drucksignal.</p>	
<p><b>Ultraschallschranke</b> bis 1,5 m. Die Pfeife links erzeugt Ultraschall. Bei freiem Durchgang verstärkt sich der Schalldruck im Empfänger, einmal durch den Trichter, und zum anderen durch den Turbulenzverstärker. Das Drucksignal <math>x</math> wird weiter in ein 0-1-Signal verstärkt.</p>	

## 4.4 Pneumatische Aktoren

Das Hauptanwendungsgebiet der Pneumatik sind heute die Aktoren. Sie sind robust, langlebig, wartungsarm, leicht zu ersetzen.

**Der Zylinder** ist das häufigste Arbeitselement der Pneumatik. Die Kolbengeschwindigkeiten liegen zwischen 0,05 bis 2 m/s. Der verwendete Luftdruck liegt zwischen 4 und 8 bar. Die Durchmesser liegen zwischen 8 mm und 350 mm. Große pneumatische Zylinder heben bis zu 8 t. Endlagendämpfer aus Gummi verhindern ein hartes Aufschlagen des Kolbens, erzeugen aber Schwingungen. Die dynamische Kraft eines Zylinders während des Ausfahrens oder Einfahrens beträgt ca 80% der statischen Kraft  $F=P \cdot A$  (Druck mal Querschnitt). Dieser Kraftverlust beruht auf dem Druckabfall in den Zuluftleitungen und einer Druckzunahme in den Abluftleitungen während die Luft strömt. Der Wirkungsgrad von Zylindern liegt bei  $\eta=0,5$ .

<b>Einfach wirkende Zylinder</b> benötigen nur eine Arbeitsleitung, dafür aber eine Rückstellfeder	<b>Doppelt wirkende Zylinder</b> fahren mit Druck ein und aus. Die Feder wirkt bei Druckausfall.
	

Es gibt zahlreiche **Sonderbauarten von Zylindern** (eine Auswahl):

- 3- und 4-Stellungszylinder statt mit nur zwei Arbeitsstellungen
- Tandemzylinder: (mehrere parallel wirkende Zylinder vereinen ihre Kraft. Der Vorteil ist die geringe Bauhöhe des einzelnen Zylinders)
- Schlagzylinder (Presslufthammer, Niethammer, Schmiedehammer)
- Seilzylinder (Ein flaches, flexibles Metallband wird durch kleine Rollen so geführt, dass es außen am Zylindermantel anliegt. Es ist durch luftdichte Schlitze mit der Bewegung des Kolbens im Zylinder verbunden. Man erreicht dadurch eine extrem kurze Baulänge.)
- Drehzylinder (eine am Kolben befestigte Zahnstange erzeugt die Drehung)
- Drehflügelzylinder (kann bis 3/4 Drehung ausführen, z.B. bei Ventilkappen eingesetzt)
- Membranzylinder (werden sehr oft als Stellantrieb bei Ventilen verwendet. Die Membran ist gummiertes Leinen zwischen zwei Stahlplatten, deren Durchmesser etwas kleiner ist, als der des Zylinders.. Die Stahlplatten nehmen den Druck der Feder oder der Federn auf. Die Bewegung beträgt meist nur wenige Millimeter oder Zentimeter)

**Druckluftmotore** arbeiten wie umgekehrte Kompressoren. Mit Druckluft werden die Kolben bewegt. Die Kurbelwelle setzt diese Bewegung in Drehbewegung um. Der Wirkungsgrad ist etwa  $\eta=0,1$ . Man baut sie bis zu Leistungen von 100 KW. **Luftdruckturbinen** haben meist eine geringe Leistung (ca. 1 W), erreichen dafür aber 350.000 U/min (Zahnbohrer).

**Düsengeräte** werden beim Versprühen von Farbe, zum Ausblasen von Spänen und zum Sandstrahlen benutzt.

**Der Schwebetisch** besteht aus zwei schweren planen Metallplatten, zwischen die durch Düsen in der unteren Platte Luft gepresst wird. Die obere Platte schwimmt auf einem Luftpolster von wenigen Zehntel mm auf und kann Lasten bis zu 300 t tragen. Diese Lasten lassen sich relativ mühelos in eine gewünschte Arbeitsposition verschieben. Stellt man die Luft ab, setzt sich die obere Platte und das schwere Bauteil wird in seiner neuen Position fixiert.

**Pneumatikfinger** gibt es in unterschiedlichen Ausführungen. Sauger heben z.B. mit Unterdruck Papierbögen von einem Stapel an, um sie einer Druckmaschine zuzuführen. Bestimmte Arten von **Roboterfingern** bestehen z.B. aus luftdichten Nachbildungen von Händen. Diese "Handschuhe" sind neben den Eisenfingern des Roboters zusätzlich mit kleinen Kugeln gefüllt, die einen weichen Griff des Roboters bewirken. Saugt man die Luft aus dem Handschuh, werden die kleinen Kugeln starr aneinander gepresst, und der Handschuh erstarrt. Der Roboter kann jetzt ein empfindliches Elektronikteil heben, ohne dass es aus der "Hand" fällt und ohne dass es zerbricht.

## 5. Vertiefung Regelungstechnik

Die Vorlesung Regelungstechnik soll in diesem Kapitel weiter vertieft werden. Das Kapitel *Regeln mit SPS* behandelt hauptsächlich die Verwendung von analogen Messwerten und den Aufbau eines Regelkreises in der SPS. *Fuzzy-Control* wendet man bei schwierigen Regelungsaufgaben an, bei denen Merkmale gekoppelt sind oder sich stark nichtlinear verhalten. Die *Neuronalen Netze* sind Neuland für die Regelung, aber erste Schritte wurden bereits getan. Das Kapitel *Mehrgrößenregelung* behandelt einen einfachen Fall gekoppelter Regelkreise nach der klassischen Methode der Entkopplungsregler.

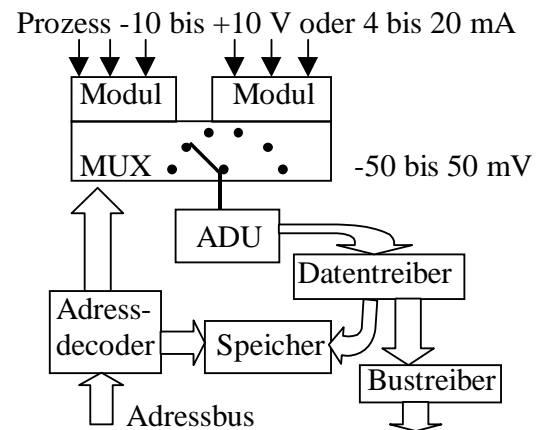
### 5.1 Regeln mit der SPS (Beispielhaft für die S5-Serie aus dem AT-Praktikum)

Wie man mit einer SPS steuert, wird in der Vorlesung Regelungstechnik behandelt. Steuern heißt hier, binäre Signale auszuwerten und mit den Mitteln der Logik und anderen binären Schaltelementen (Flipflop, Timer, Zähler) Prozessschritte zu starten und zu beenden, und innerhalb der Prozessschritte die richtigen Sensoren zu befragen und die richtigen Aktoren zu stellen.

#### 5.1.1 Lesen und Wandeln von Analogsignalen allgemein

Analogsignale sind z.B. Stromsignale im Bereich 4-20 mA (Industrienorm) oder Spannungssignale im Bereich -10 bis +10 V (Industrienorm). Beim Einlesen in den Rechner werden die Signale in Zahlen umgewandelt.

Vom Prozess kommen analoge Signale. Ein **Modul** wandelt sie in Spannungen (-50 bis +50 mV). Der **Multiplexer (MUX)** ist ein Kanalschalter, der vom Adressbus gesteuert wird. Er wählt einen Messkanal und leitet den Wert an den **Analog-Digital-Umwandler (ADU)**. Der erzeugt ein 16-Bit- oder 32-Bit-Zahlenwort mit dem Wert der analogen Größe. Der Datentreiber bringt das Wort zum Speicher, wo für jeden Kanal ein Speicherwort für eine Zwischenspeicherung bereit steht, und auch zum Bustreiber, falls das SPS-Programm gerade diesen Wert haben möchte. Die Kanaladresse wird im Adressdecoder entschlüsselt.



- Ein **Kanal** ist die Nummer einer Signalklemme für eine Messleitung, meist pro Baugruppe von 0 bis 7 nummeriert
- Eine **Baugruppe** hat meist 8 Kanäle. Die **Baugruppennummer** ist eigentlich eine Speicheradresse der SPS im Portbereich für eine 8-er-Gruppe von Speicherworten. Der Portbereich ist ein Speicherbereich, auf den sowohl externe Geräte (z.B. der Bustreiber der Baugruppe) als auch die CPU (Central Processing Unit, d.h. der eigentliche  $\mu$ -Controller in der SPS) zugreifen können, vergleichbar einem Bücherregal, das von beiden Seiten begehbar ist.
- Eine Messung heißt **potentialgetrennt**, wenn keine der beiden Zuleitungen geerdet ist. Messgröße ist dann die Differenzspannung. Jede Strippe wird gegen Erde gemessen und dann rechnerisch intern die Differenz gebildet, d.h. beide Messleitungen müssen sich nach dem Anpassungsmodul im Messbereich -50 bis +50 mV bewegen.
- Eine Messung heißt **potentialgebunden**, wenn nur eine spannungs- oder stromführende Messleitung benutzt wird und die zweite Leitung an der Sammelerde der Analogeingabe angeklemmt ist.
- Ein **Shunt** (eine elektrische Hilfsverbindung, d.h. ein definierter Widerstand zwischen Messleitung und Erde, meist als Spannungsteiler aus zwei Widerständen aufgebaut) wandelt bei Stromeingängen das Stromsignal in das geforderte Spannungssignal im Bereich 0-50 mV für den Wandler (ADU).
- **Drahtbruchmeldung** ist ein SPS-Service, der nur bei Stromeingängen angeboten wird. Vor jeder Messung wird 1,6 ms lang ein Konstantstrom auf die Messleitung aufgegeben. Der angeschlossene Messumformer muss das akzeptieren können, ohne Schaden zu nehmen. Steigt dabei die Spannung in der Messleitung auf den höchsten erreichbaren Wert des Konstantstromgebers (ca. 12 Volt), dann nimmt man Drahtbruch an, und eine entsprechende Fehlermeldung wird erzeugt. Zusätzlich muss man in der Baugruppe für diesen Kanal einen Jumper stecken (Jumper sind kleine Schalter oder Kurzschlussstecker, mit denen man bestimmte Bedingungen setzen kann. Die Elektronik fragt die Schalterstellung dann ab)
- Als **Abtastung** bezeichnet man das Messen eines Messwertes. Für die Regelung ist eine zyklische Abtastung in gleichen Zeitabständen wichtig (**Tastzeit TA**). Werden alle 8 Kanäle einer Baugruppe zyklisch abgetastet, dann ist bei einer Wandelzeit von ca. 60 ms/Kanal ein bestimmter Kanal nur alle 480 ms dran. Das ist etwa alle halbe Sekunde.
- **Einzelabtastung** eines Kanals heißt, dass der Multiplexer nur diesen einen Kanal wählt und der ADU nur diesen Kanal wandelt. Auf diese Weise kann man die 8-fache Abtastrate erreichen.

### 5.1.2 Die Programmiersprache AWL (Anweisungsliste)

Viele SPS-Programme werden auch heute noch in der Maschinensprache geschrieben. Das trifft insbesondere auf kompliziertere Kommunikationsprogramme, aber auch auf die Analogwertverarbeitung zu. Eine Maschinensprache verwendet abgekürzte Bezeichnungen für die Befehle und für die Daten. Ein Befehl macht immer nur sehr wenig, so dass ein Programmstück aus sehr vielen einzelnen Befehlen bestehen kann.

Liste einiger Befehle und Konstanten:

Beschreibung des Befehls oder der Konstanten	Beispiel
Sprung absolut zu einem Programmbaustein, z.B. PB1	SPA PB 1
Sprung absolut zu einer Zielmarke, z.B. zur Marke M007	SPA =M007
Bedingter Sprung zu einer Zielmarke. Dieser Sprung wird nur ausgeführt, wenn im 1-Bit-Vergleicherregister (VG) eine 1 steht	SPB =M007
Aufruf eines Datenblocks, z.B. DB 10. Die SPS merkt sich diese Nummer in einem Register. DB 10 wird zum <i>aktiven</i> Datenblock	A DB 10
Laden eines 2-Byte-Wortes (Wort 22 z.B.) aus dem aktiven Datenblock in das Akkumulatorregister Akku1 (der alte Inhalt wandert vorher nach Akku2. Dessen alter Inhalt verschwindet)	L DW 22
Laden eines Bytes vom Merkerspeicher nach Akku1	L MB 488
Transferieren (Umspeichern) des Akku1-Inhalts in den Speicher	T MB 555
Größenvergleich der beiden Akkuinhalte, z.B. Akku2 > Akku1 Das Ergebnis erscheint im Vergleicher (Ja: VG=1, Nein: VG=0)	F>
Laden eines Merkerbits in den Vergleicher. Das kann mit einem UND-Befehl oder mit einem ODER-Befehl geschehen	U M 25.3 O M 25.3
Negiertes Laden eines Merkerbits (z.B. eine 0 wird als 1 geladen)	UN M 25.3
Setzen einer 1 auf ein Merkerbit wenn der Vergleicher VG=1 ist	S M 25.3
Setzen einer 0 auf ein Merkerbit wenn der Vergleicher VG=1 ist	R M 25.3
Abspeichern des Vergleicherinhalts auf ein Merkerbit	= M 25.3
Festkommakonstante, z.B. eine Baugruppenadresse	KF +192
Doppelkonstante (2 Zahlen zu je einem Byte Speicherplatz)	KY 4, 7
Bitmaskenkonstante mit einer Bitbelegung eines 16-Bit-Wortes	KM 00110011 11110000

### 5.1.3 Datenbaustein für die PID-Regelung (bei der S5-Serie von SPSen benötigt)

Für jeden Regelkreis benötigt man einen eigenen Datenbaustein. Datenbausteine werden nummeriert, z.B. DB 1, DB 2, ..., DB 255. Der Datenbaustein einer PID-Regelung umfasst genau 49 Datenworte a 16 Bit (2 Byte). In diesem Datenbaustein werden alle wichtigen Informationen zusammengetragen, damit der PID-Algorithmus (vom Hersteller der SPS mitgeliefert) arbeiten kann. Der Datenbaustein wird vom Programmierer als *neuer Baustein* angelegt und dann ausgefüllt. Im laufenden Betrieb ändern sich dann nur noch der Sollwert W, die Istgröße x und die Stellgröße y. Die Taktzeit TA taucht hier nur indirekt bei der Berechnung von KI und KD auf.

Die wichtigsten Eintragungen im PID-Baustein der S5-Serie sind:

Wort	Inhalt	Beschreibung
DW1	K	Reglerverstärkung, z.B. KF +17053 bei $K_{PR}=17,053$ (Faktor 1000)
DW3	R	Zusatzfaktor im P-Anteil. Die Gesamtverstärkung ist hier $K \cdot R$ (Faktor 1000)
DW5	TI	=TA/TN Integrierkonstante, z.B. KF +44 bei TI=43,59 (Faktor 1000)
DW7	TD	=TV/TA Differenzierkonstante, z.B. KF +7 bei TD=6,51 (Faktor 1)
DW9	W	Sollwert im Zahlenbereich von -2047 bis +2047
DW11	STEU	Steuerwort mit Steuerinformation (siehe unten)
DW12	yH	Handwert der Stellgröße y bei Umschaltung in Handbetrieb
DW14	BGOG	Obere Begrenzung der Stellgröße y im Zahlenbereich von -2047 bis +2047
DW16	BGUG	Untere Begrenzung der Stellgröße y im Zahlenbereich von -2047 bis +2047
DW22	X	Istwert der Regelgröße im Zahlenbereich von -2047 bis +2047
DW24	Z	Hauptstörgröße bei Aufschaltung im Zahlenbereich von -2047 bis +2047
DW48	y	Ausgabewert der Stellgröße y im Zahlenbereich von -2047 bis +2047

Das **Steuerwort** besteht aus 2 Bytes. Das high Byte wird vom Programmierer mit Nullen belegt und darf im Betrieb nur vom PID-Baustein selbst verändert werden. Die Bits 0 bis 5 des low Byte haben jeweils eine Steuerfunktion, z.B.

Bit0 = 0 heißt Handbetrieb	Bit0 = 1 heißt Automatikbetrieb
Bit2 = 0 heißt Regelbaustein arbeitet	Bit2 = 1 heißt Regelbaustein arbeitet nicht
Bit3 = 0 heißt K-Regler	Bit3 = 1 heißt S-Regler
Bit5 = 0 heißt ohne Aufschaltung	Bit5 = 1 heißt mit Aufschaltung

#### 5.1.4 Lesen von Analogsignalen mit FB 250 (bei der S5-Serie von SPS-en)

Es ist nicht üblich, die gewandelten Zahlenwerte direkt aus den Portspeichern zu lesen. Dafür gibt es einige Gründe. Der AWL-Programmierer bedient sich viel mehr eines vom Hersteller mitgelieferten Programms (Funktionsbaustein FB 250 bei der S5-Serie).

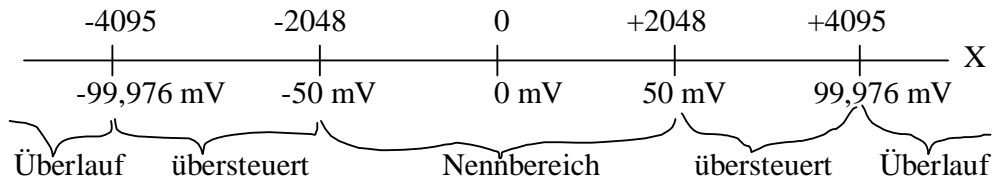
Das Eingabewort der SPS hat 16-Bit und den folgenden Aufbau:

Byteadressen	n								n+1							
Bitnummern im Byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bedeutung der Bits	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	T	F	Ü
Bedeutung des Bytes	high Byte								low Byte							

Die Bits  $2^{12}$  bis  $2^0$  enthalten den gewandelten Zahlenwert in binärer Form. Sie sind mit Null oder Eins belegt je nach Zahlenwert. Der Zahlenwert kann auch negativ sein (siehe 2-er-Komplement). Dabei enthält das *high Byte* die höherwertigen Dualstellen, das *low Byte* die niederwertigen. Die Byteadresse des high Bytes ist jedoch niedriger als die des low Byte. Beide Bytes liegen im Speicher unmittelbar hintereinander. Die drei letzten Bits (T, F, Ü) im low Byte sind für Eingabefehlermeldungen reserviert. Tritt kein Fehler auf, sind sie 0.

Bitbelegung T=1 meldet Tätigkeit des Wandlers (z.B. Einzelabtastung für einen anderen Kanal)  
 Bitbelegung F=1 meldet einen Drahtbruch (nur bei Stromeingang und gesetztem Jumper)  
 Bitbelegung Ü=1 meldet Überlauf. Der Wert der analogen Größe ist zu groß.

Intern wird die analoge Größe auf eine Zahlengerade X abgebildet:



Die Ausdehnung des Nennbereichs kann jedoch durch die Angabe neuer Grenzen noch abgeändert werden, z.B. so, dass der positive Bereich von 0 bis 50 mV auf die Zahlen 0 bis 1000 abgebildet wird. Einen Merker mit ständiger Bitbelegung 0 bezeichnet man als **Nullmerker**.

Sobald der Programmierer am Programmiergerät den Befehl "**SPA FB 250**" eintippt und die ENTER-Taste betätigt, erscheint nach dem Befehl eine farbige Maske zum Ausfüllen (fette Schrift markiert die Maske):

```

: SPA FB 250 Aufruf der Analogeingabe, z.B. eines Sollwertes
Name : RLG:AE Diese Zeile sollten Sie nicht ändern
BG : KF +192 Baugruppe
KNKT: KY 0, 4 Kanalnummer, Kanaltyp (Der Kanaltyp ist codiert)
OGR : KF +1000 Obere Grenze Nennbereichsumrechnung (2048—>1000)
UGR : KF +0 Untere Grenze Nennbereichsumrechnung (0 —> 0)
EINZ : M 12.0 Falls M 12.0 ein Nullmerker ist, dann keine Einzelabtastung
XA : DW 9 Zielort der Eingabe ist hier Datenwort 9 im aktuellen DB
FB : M 25.0 Zielmerker für Drahtbruchmeldung ist hier Merker 25.0
BU : M 25.1 Zielmerker für Meldung Bereichsüberschreitung
TBIT : M 25.2 Zielmerker für Meldung ADU anderweitig beschäftigt

```

### Einige Kanaltypen für die Eingabe

Typ	Beschreibung
3	Stromeingabe 4 bis 20 mA
4	Spannungseingabe unipolar, d.h. nur positive Spannungen von 0 bis 10 V
6	Spannungseingabe bipolar, d.h. alle Spannungen von -10 bis +10 V

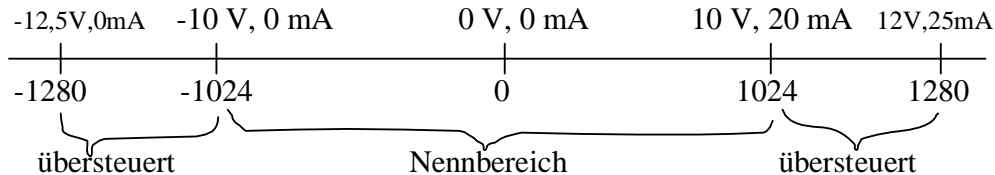
#### 5.1.5 Ausgeben von Analogsignalen mit FB 251 (bei der S5-Serie von SPS-en)

Eine Analogausgabe benötigt man z.B., um den Hub eines Regelventils einzustellen oder die Rührerdrehzahl im Fermenter einzustellen.

Das Ausgabewort der SPS hat ebenfalls 16-Bit und den folgenden Aufbau:

Byteadressen	n	n+1
Bitnummern im Byte	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Bedeutung der Bits	$2^{11}$ $2^{10}$ $2^9$ $2^8$ $2^7$ $2^6$ $2^5$ $2^4$	$2^3$ $2^2$ $2^1$ $2^0$ <b>X</b> <b>X</b> <b>X</b> <b>X</b>
Bedeutung des Bytes	high Byte	low Byte

Die Bits  $2^{11}$  bis  $2^0$  enthalten den Zahlenwert in binärer Form, der als Analogwert, z.B. als Strom- oder Spannungssignal ausgegeben werden soll. Die 4 mit X markierten Bits sind beliebig, jedoch belegen sie die meisten Programmierer mit Nullen. Um diese Details bemüht sich jedoch der Funktionsbaustein FB 251. Extern wird die Zahlengerade X entweder auf einen Stromwert oder auf einen Spannungswert abgebildet (es gibt jedoch keine negativen Ströme):



Sobald der Programmierer am Programmiergerät den Befehl "**SPA FB 251**" eintippt und die ENTER-Taste betätigt, erscheint nach dem Befehl eine farbige Maske zum Ausfüllen (fette Schrift markiert die Maske):

	: SPA FB	251	Aufruf der Analogausgabe, z.B. Stellsignal für ein Ventil
<b>Name</b>	: <b>RLR:AA</b>		Diese Zeile sollten Sie nicht ändern
<b>XE</b>	: DW	48	Herkunft der auszugebenden Zahl (hier aus aktuellem DB)
<b>BG</b>	: <b>KF</b>	+224	Baugruppe
<b>KNKT</b>	: <b>KY</b>	0, 0	Kanalnummer, Kanaltyp (Der Kanaltyp ist codiert)
<b>OGR</b>	: <b>KF</b>	+1000	Obere Grenze Nennbereichsumrechnung (1000 → 1024)
<b>UGR</b>	: <b>KF</b>	+0	Untere Grenze Nennbereichsumrechnung (0 → 0)
<b>FEH</b>	: <b>M</b>	26.0	Zielmerker für Fehlermeldung $UGR \geq OGR$ ist hier M 26.0
<b>BU</b>	: <b>M</b>	26.1	Zielmerker <i>Bereichsüberschreitung</i> $XE < UGR, XE > OGR$

OGR und UGR dürfen hier nicht willkürlich gesetzt werden, wie bei der Analogeingabe, sondern müssen identisch sein mit den Grenzvorgaben für die Stellgröße im Datenblock des Regelkreises.

**Achtung !** An einem Spannungsausgang müssen mindestens **3,3 K $\Omega$**  Widerstand hängen, da sonst die Stromstärke zu hoch wird  
 An einem Stromausgang dürfen maximal **300  $\Omega$**  Widerstand hängen, da sonst die erforderliche Spannung zu hoch wird

Kanaltypen der Ausgabe:

Z.B. ist Kanaltyp 0 ein unipolarer Spannungsausgang mit dem Ausgabebereich 0 bis 10 Volt

Z.B. ist Kanaltyp 1 ein bipolarer Spannungsausgang mit dem Ausgabebereich -10 bis 10 Volt

### 5.1.6 Der PID-Baustein OB251 (bei den S5-Serien)

Für die Regelung wichtig ist eine konstante Tastzeit TA, die beim Start des Programms festgelegt wird. Aus der Gleichung des idealen PID-Reglers mit den Bestandteilen **P**roportional, **I**ntegrierend, **D**ifferenzierend

$$y = K_{PR} \left[ R \cdot xd + \frac{1}{T_N} \int_0^t xd(u) du + T_V \dot{xd} \right]$$

mit  $xd = W - x$  ergibt sich nach einer Differenziation

$$\dot{y} = K_{PR} \left[ R \cdot \dot{xd} + \frac{1}{T_N} xd + T_V \ddot{xd} \right]$$

und mit näherungsweise

$$\dot{y}_n = \frac{y_n - y_{n-1}}{TA}, \quad \dot{xd}_n = \frac{xd_n - xd_{n-1}}{TA} \quad \text{und} \quad \ddot{xd}_n = \frac{\dot{xd}_n - \dot{xd}_{n-1}}{TA} = \frac{xd_n - 2xd_{n-1} + xd_{n-2}}{TA^2}$$

die Differenzgleichung des idealen PID-Algorithmus (DIN-Algorithmus)

$$y_n = y_{n-1} + K_{PR} \left[ R (xd_n - xd_{n-1}) + \frac{TA}{T_N} xd_n + \frac{T_V}{TA} (xd_n - 2xd_{n-1} + xd_{n-2}) \right]$$

Hierbei ist:

- n eine fiktive Nummerierung 1, 2, 3, ... der Tastschritte, die im Rechner nicht wirklich erfolgt
- $y_n$  der Stellgrößenwert nach der n-ten Abtastung berechnet
- $K_{PR}$  der Teil der Reglerverstärkung, der auf alle 3 PID-Anteile wirken soll (P-, I-, D-Anteil)
- R der Teil der Reglerverstärkung, der zusätzlich den P-Anteil steuern soll
- $xd_n = W - x$  die Regeldifferenz Sollwert minus Istwert im n-ten Tastschritt
- $xd_{n-1}$  die Regeldifferenz des vorangehenden Tastschritts
- $xd_{n-2}$  die Regeldifferenz zwei Tastschritte davor
- $T_N$  die Nachstellzeit des PID-Reglers mit Faktor  $K_{PR}TA/T_N$  als Faktor des I-Anteils
- $T_V$  die Vorhaltezeit des PID-Reglers mit Faktor  $K_{PR}T_V/TA$  als Faktor des D-Anteils
- Der I-Anteil wird 0, wenn Faktor  $TA/T_N$  Null gesetzt wird
- Der D-Anteil wird 0, wenn Faktor  $T_V/TA$  Null gesetzt wird

Ein **S-Regler** (Schritt- oder Geschwindigkeitsalgorithmus) gibt nicht  $y_n$  aus, sondern die Differenz  $\Delta y = y_n - y_{n-1}$ . Die Summation erfolgt im Stellglied. Beispiel sei ein Regelventil, das durch einen Motor weiter auf- bzw. weiter zu gemacht werden soll. Ein positives  $\Delta y$  bewirkt Rechtslauf des Motors und damit ein Auffahren des Ventils, ein negatives  $\Delta y$  hingegen Linkslauf und Zufahren. Der Betrag  $|\Delta y|$  ist der Drehzahl des Motors proportional. Der Regelkreis benötigt eine Rückmeldung über die genaue Stellung des Stellgliedes, aber keinen Stellungsregler.

Ein **K-Regler** (kontinuierlicher Regler) gibt  $y_n$  aus. Die genaue Einstellung des Stellgliedes übernimmt ein Stellungsregler im Stellantrieb. Dafür spart man die Rückmeldung über die Stellung des Stellgliedes ein, die beim S-Regler notwendig ist.

**Der PID-Algorithmus ist als Organisationsbaustein OB 251** (S5-Serie) bereits in das Betriebssystem der SPS integriert. Man ruft ihn einfach auf. Der Aufruf muss jedoch zyklisch zeitgesteuert erfolgen, damit die Tastszeit TA peinlich genau eingehalten wird. Der Aufruf wird durch eine in die SPS integrierte Uhr ausgelöst, deren Zeitimpulse einen Interrupt in der CPU auslösen. Ist die Tastszeit eines Regelkreises abgelaufen, wird z.B. Organisationsbaustein OB13 aufgerufen, der dann die eigentliche PID-Regelung anstößt (Siehe Organigramm unten).

4 verschiedene **Tastzeiten** sind möglich: TA<sub>1</sub>, TA<sub>2</sub>, TA<sub>3</sub> und TA<sub>4</sub>. Ist TA<sub>1</sub> abgelaufen, wird OB13 per Interrupt aufgerufen, ist TA<sub>2</sub> abgelaufen, dann OB12, ist TA<sub>3</sub> abgelaufen, dann OB11, ist TA<sub>4</sub> abgelaufen, dann OB10. Laufen mehrere Tastzeiten gleichzeitig ab, gilt eine Vorrangregelung. Wozu braucht man unterschiedliche Tastzeiten? Z.B. ändert sich die Temperatur in einem großen Fermenter nur langsam. Hier reichen Tastzeiten von 10 Sekunden und mehr vollkommen aus für eine Regelung. Ist es ein Membranreaktor, dann kann sich jedoch der Membrandruck in Minuten dramatisch ändern, und man regelt die Pumpenleistung besser mit einer kürzeren Tastzeit, z.B. 1 Sekunde.

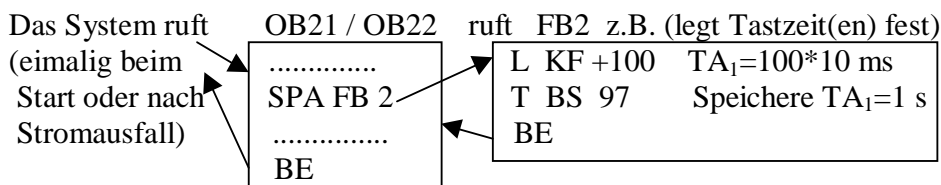
Der **ideale** und der **reale PID-Regler** (letzterer auch **PIDT<sub>1</sub>**-Regler genannt) unterscheiden sich in der Behandlung des differenzierenden Anteils (D-Anteil). Der reale PID-Regler hat einen zusätzlichen Reglerparameter **VV** (Vorhalteverstärkung) bzw. **T<sub>1</sub>** (Zeitkonstante des D-Anteils) je nach Hersteller. Die beiden Größen sind über die Formel  $VV = T_V / T_1$  verknüpft. Die DGL des realen PID-Reglers enthält auf der linken Seite zusätzlich die Ableitung des D-Anteils nach der Zeit:

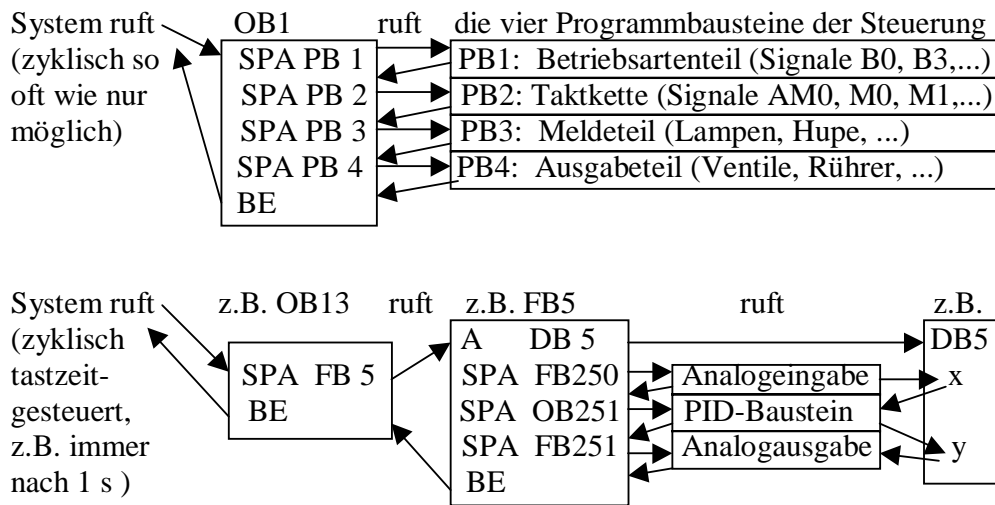
$$T_1 \dot{y}_D + y = K_{PR} \left[ R \cdot xd + \frac{1}{T_N} \int_0^t xd(u) du + T_V \dot{xd} \right]$$

Den Unterschied in der Wirkung zeigen die folgenden kleinen Graphiken:

	<p>Eine Sollwertrampe <math>W(t)</math> führt über die Gleichung <math>xd = W - x</math> zu einer Rampe der Regeldifferenz <math>xd</math>, wenn die Regelung nur langsam genug reagiert. Die Punkte stellen die Tastzeitpunkte dar, an denen der Istwert <math>x</math> gemessen wird.</p>
	<p>Der ideale PID-Regler erzeugt nur an den Knickstellen der <math>xd</math>-Kurve <math>y_D</math>-Impulse (die aber von beträchtlicher Größe), denn nur dort ist eine Krümmung (zweite Ableitung) der <math>xd</math>-Kurve vorhanden. Der Gesamt-<math>y</math>-Wert kann auf diese Weise negativ werden oder über 100% klettern, so dass die Wirkung verpufft.</p>
	<p>Der reale PID-Regler verteilt die starken <math>y_D</math>-Impulse auf eine Treppe von Impulsen mit abnehmender Höhe, aber derselben Fläche. Dadurch hält die Wirkung länger an, und die Gefahr sinkt, dass das Gesamt-<math>y</math> negativ wird oder über 100% steigt.</p>

**Organigramm eines SPS-Steuer- und Regelprogramms (S5-Serie aus dem AT-Praktikum)**





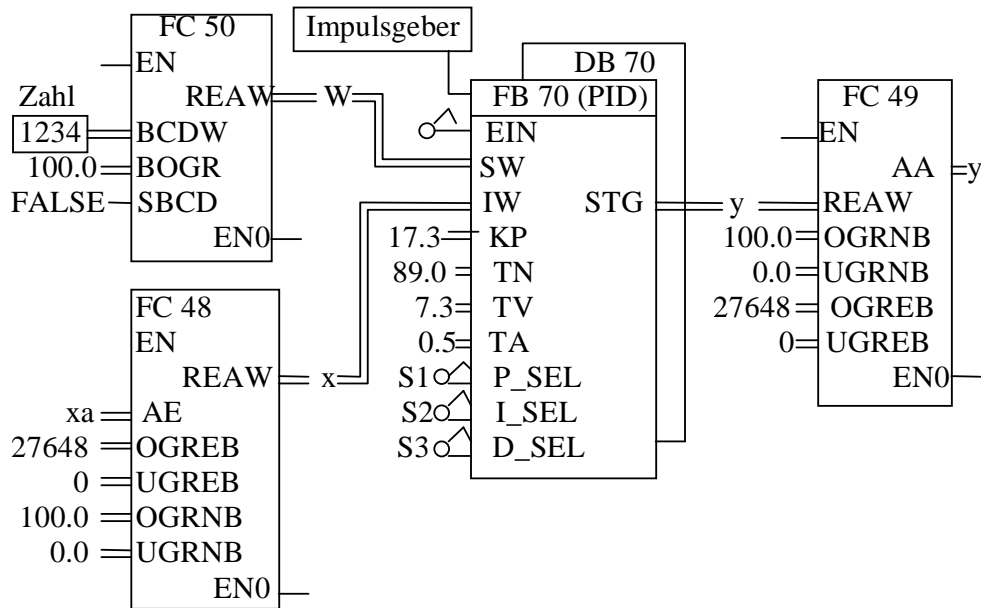
### 5.1.7 Graphische Reglerprogrammierung (Ausblick auf neuere SPS)

Die S5-Serie ist veraltet und wird im Skript nur deshalb so ausführlich behandelt, weil

1. sie im AT-Praktikum von UV/BT immer noch verwendet wird
2. die Grundprinzipien sich kaum geändert haben

Neuere SPSen haben Gleitkommaprozessoren, wesentlich mehr Programm- und Datenspeicher, eine höhere Geschwindigkeit und bessere Kommunikationsmöglichkeiten. Augenfällig in der Programmierung ist, dass viele Standard-Funktionsbausteine neben ihrer AWL-Aufrufmaske auch eine graphische Aufruf-Repräsentation in Boxenform haben. Mit diesen Boxen können Datenflusspläne (Regelungsprogramme) ähnlich den Funktionsplänen der digitalen Steuerung gezeichnet werden. Für viele Standardaufgaben reicht diese graphische Programmiermethode vollkommen aus. Zudem zwingt sie zur Verwendung vorgefertigter ausgetesteter Funktionsbausteine und vermindert so die Zahl der Programmierfehler. Für tieferegehende Studien oder praktische Anwendungen wird die Literatur empfohlen, z.B. Wellenreuther / Zastrow. Gegenüber der S5-Serie sind z.B. bei der S7-Serie unter anderem neue Bausteinnummern eingeführt worden und z.T. neue Datenblockkonzepte.

Die folgende Graphik (an Wellenreuther / Zastrow angelehnt) zeigt die Eingabe eines Sollwertes über eine Zifferneingabetastatur und die Wandlung der Dezimalzahl mit der Funktion FC 50 in eine REAL-Zahl (Gleitkommaparstellung intern), die Analogeingabe des Istwertes mit Funktion FC 48 und die Normierung der gelesenen REAL-Zahl auf den Bereich 0 bis 100, den Aufruf des PID-Bausteins (FB 70), und die Stellgrößenausgabe mit Funktion FC 49. Neu ist z.B. die Vorgabe der Tastzeit TA zusammen mit den Reglerparametern. Für die Speicherung solcher unveränderlichen Konstanten benutzt die S7-Philosophie sogenannte Instanz-Datenbausteine. Jede Reglerbox ist z.B. eine Instanz (eine Kopie) der allgemeinen PID-Aufrufmaske und hat einen eigenen Konstantenspeicher, eben diesen Instanz-Datenbaustein. Zusätzlich sind aber auch allgemein zugängliche Datenbausteine möglich, wie wir sie von der S5-Serie her kennen.



In der Graphik bedeutet:

REAW	ein Datenwort vom Typ REAL
BCDW	ein Datenwort im BCD-Code (einzeln verschlüsselte Dezimalziffern)
BOGR	obere Begrenzung der gewandelten Zahl
FALSE	logischer Wert "false" (z.B. ein Nullmerker)
SBCD	Selektor BCD, d.h. sind auch die Pseudoziffern A, B, ..., F erlaubt?
DB 70	Der Instanzdatenblock zu gerade diesem PID-Aufruf
SW, IW, KP, TN	Sollwert, Istwert, Reglervverstärkung, Regler-Nachstellzeit
TV, TA	Regler-Vorhaltezeit, Tastzeit
S1, P_SEL	Schalter 1, Selektor für den P-Anteil aus der Stellgröße $y=P+I+D$
OGRNB, UGRNB	Obere und untere Begrenzung des Nennbereichs
OGREB, UGREB	Obere und untere Begrenzung der internen Festkommdarstellung von y

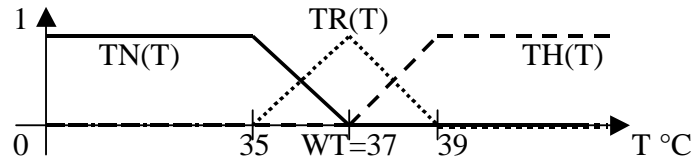
Für die Anlagenumrüstung, z.B. von der veralteten S5-Serie auf die modernere S7-Serie stehen **Interface-Module** zur Verfügung, die es ermöglichen, einzelne SPS-en ohne großen Verdrahtungsaufwand und Stillstand auszutauschen.

## 5.2 Fuzzy Control

Lotfi Zadeh führte 1965 den Begriff der unscharfen Logik ein (Fuzzy Logic). Statt **wahr** und **falsch** sind jetzt auch Grautöne zugelassen. Meist wird false=0 und wahr=1 gesetzt. Die Fuzzy Logic rechnet demnach mit allen Zahlen aus dem Intervall [0 , 1]. Das folgende Beispiel orientiert sich an der gemeinsamen Regelung von Temperatur **T** und O<sub>2</sub>-Konzentration **O**. Wir stellen uns eine kleine Graphik auf dem Bildschirm des Leitstandes vor:



durchgezogen,  $TR(T)$  gepunktet und  $TH(T)$  gestrichelt. Wie man sieht, liefert die Funktion **TR**ichtig(T) nur den Wert 1, wenn die Temperatur genau im Sollwert  $37^\circ\text{C}$  ist. Die Funktion **TH**och dagegen liefert den Wert 1 dagegen erst, wenn die Temperatur  $T$  über  $39^\circ\text{C}$  steigt. Zwischen  $37$  und  $39^\circ\text{C}$  ist die Temperatur *mehr oder weniger hoch*, d.h., es wird ein Wert zwischen 0 und 1 geliefert.



In ähnlicher Weise werden alle 12 Zuordnungsfunktionen programmiert. Natürlich sind auch andere Funktionsformen als die dargestellten Rampen möglich, nur sollte die Summe der Zuordnungsfunktionen zu einer Variablen, z.B.  $TN(T)+TR(T)+TH(T)$  zur Variablen  $T$ , immer möglichst genau den Wert 1 ergeben. Hier bei den Rampen ist diese Bedingung immer exakt erfüllt.

Beispiel:  $T=35,64^\circ\text{C}$   
 Solltemperatur  $W_T=37^\circ\text{C}$   
 $TN(T)=(37 - 35,64)/2= 0,68$  d.h., zu etwa  $2/3$  niedrig  
 $TR(T)=(35,64 - 35)/2= 0,32$  d.h., zu  $1/3$  richtig  
 $TH(T)=0$  d.h., kein bisschen zu hoch  
 $TN+TR+TH=1$

## Inference

Inference (oder Schlussfolgerung) heißt die Abfrage von **Expertenregeln** aus einer **Wissensbasis** nach einer bestimmten Methode, z.B. der **Min-Max-Methode**. In diesem Schritt bekommen die **Ergebnisfunktionen** ihre Werte.

Expertenregeln gewinnt man aus der Befragung von Experten. So sagt z.B. Meister Schröder in einer bestimmten Regelsituation: "Wenn die Temperatur **Niedrig** **UND** die Temperaturänderungsgeschwindigkeit **DT** Unerheblich ist, dann musst du **HeiZen**."

Es gibt zahlreiche Rechenregeln für die unscharfe Logik, die mehr oder weniger kompliziert sind. Die Min-Max-Methode z.B. ersetzt die zweiwertige dyadische logische Operation *Konjunktion* ( $a \wedge b$  bzw. "a UND b") durch die Funktion Minimum(a, b), die *Disjunktion* ( $a \vee b$  bzw. Alternative, Adjunktion, inklusives "a ODER b") durch die Funktion Maximum, und die monadische zweiwertige logische Operation *Negation* ( $\neg a$ ) durch die Funktion "Rest zu Eins". Mit false=0 und true=1 ist

Funktion	scharfe Logik				unscharfe Logik
a UND b	$1 \wedge 1 = 1$	$1 \wedge 0 = 0$	$0 \wedge 1 = 0$	$0 \wedge 0 = 0$	Minimum(a, b)
a ODER b	$1 \vee 1 = 1$	$1 \vee 0 = 1$	$0 \vee 1 = 1$	$0 \vee 0 = 0$	Maximum(a, b)
Nicht a	$\neg 1 = 0$		$\neg 0 = 1$		$1 - a$

Die 5 Ergebnisfunktionen in unserem Beispiel könnten sein:

HZ		für eine digitale Ansteuerung der Heizung (Ein / Aus)
RN	Rühre nicht	für eine kontinuierliche Drehzahlregelung des Rührers geeignet
RW	Rühre wenig	
RM	Rühre mittelstark	
RS	Rühre stark	

Da besonders bei Befragung mehrerer Experten zu einer Ergebnisfunktion mehrere Regeln mit unterschiedlichen Werten zu einer Ergebnisfunktion führen können, muss man die Werte irgend wie verknüpfen. Zwei landläufige Methoden sind die **Mittelung** der Werte zu einer Ergebnisfunktion oder die **Maximumbildung**, was einer ODER-Verknüpfung der Einzelwerte entspricht. Bei beiden Verfahren werden alle Ergebnisfunktionen vor dem Start der Inference auf den Wert 0 gesetzt, z.B. HZ=0, RN=0, ..., RS=0,. Bei der Mittelwertbildung muss zusätzlich noch die Anzahl n der addierten Werte 0 gesetzt werden, z.B. nHZ=0, nRN=0, ..., nRS=0.

Benutzen wir die Maximumbildung als Verknüpfung mehrerer Regeln zu einer Ergebnisfunktion und C als Programmiersprache, dann würde obige Beispielregel von Meister Schröder etwa so umgesetzt werden:

$$HZ = \text{Maximum} ( HZ, \text{Minimum} ( TN, DTU) )$$

Die Funktion Maximum realisiert hier die Verknüpfung mehrerer Regeln zum Heizen, die Funktion Minimum realisiert das UND aus Meister Schröders Heizregel. Der neue Wert von Ergebnisfunktion HZ wird wieder auf seinen alten Platz zurück gespeichert

### Defuzzification

Die Defuzzification berechnet Stellgrößen aus den Werten der Ergebnisfunktionen. Bei Schaltern reicht als Ergebnis ein Ein/Aus. Bei kontinuierlichen Größen können beliebige Werte herauskommen. Manchmal muss man sie jedoch normieren bzw. beschränken, z.B. auf ein Intervall [0, 1] oder ein Intervall [0, 100%].

Bei digitalen zweiwertigen Stellgrößen (Ein-Aus-Größen) arbeitet man zumeist mit einem willkürlich wählbaren Schwellenwert aus dem Intervall [0, 1], z.B. 0,5 in diesem Beispiel für die Schalterstellung H der Heizung.

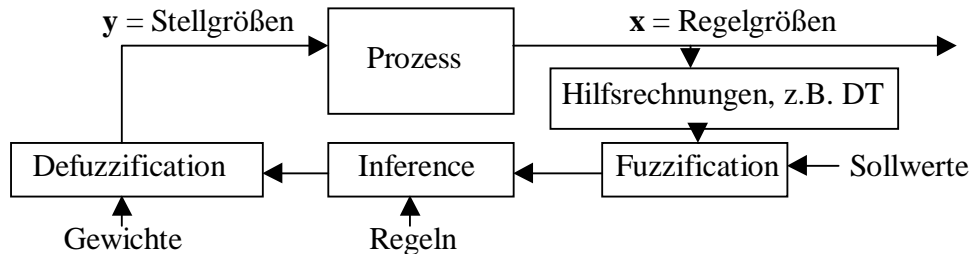
$$H = \begin{cases} 1, & \text{wenn } HZ > 0,5 \\ 0, & \text{wenn } HZ \leq 0,5 \end{cases}$$

Bei kontinuierlichen Stellgrößen ohne Normierung benutzt man z.B. die COM-Methode (Center Of Mass oder Schwerpunktmethod), wie z.B. zur Berechnung der Rührerdrehzahl R:

$$R = RN \cdot 0 + RW \cdot 30 + RM \cdot 60 + RS \cdot 120$$

Die gewählten Gewichte 0, 30, 60, 120 bestimmen den Stellgrößenverlauf. Die maximal mögliche theoretische Rührerdrehzahl ist hier 120 U/min.

## Blockschaltbild der Fuzzy Control



Wir sehen einen geschlossenen Regelkreis. Die Regelgrößen  $x$  bilden einen Vektor (da es mehrere sind), ebenso die Stellgrößen  $y$ . Die Sollwerte gehen in die Fuzzification ein und ändern dort die Zuordnungsfunktionen. Die Regeln gehen in die Inference ein. Die Gewichte gehen in die Berechnung der Ergebnisfunktionen ein. Insgesamt hat man in der Fuzzy Control mehr Möglichkeiten der Einflussnahme, als bei einer Regelung mit herkömmlichen PID-Reglern, und kann auf diese Weise auch nichtlineare und/oder gekoppelte Systeme regeln.

### Beispiel einer Hilfsrechnung, z.B. DT aus T berechnen

Zur Ausführung der eigentlichen Regelaufgaben sind oft Hilfsrechnungen notwendig. So werden z.B. Temperaturänderungsgeschwindigkeiten fast nie gemessen, sondern berechnet. DT ist die Änderungsgeschwindigkeit der Temperatur. Ist  $dt$  die Dauer eines Tastschritts, dann gilt ungefähr

$$DT = \frac{\Delta T}{dt} = \frac{T_j - T_{j-1}}{dt}$$

$T_{j-1}$  und  $T_j$  sind zwei aufeinanderfolgende Temperaturmessungen im Zeitabstand  $dt$ . Da differenzierte Kurven eine starke Neigung zur Aufrauung haben, muss man sie glätten. Das einfachste Glättungsverfahren ist die exponentielle Glättung (siehe auch Kap. 2.1 Gerätesteuerung mit  $\mu$ -Controller). So werden z.B. vom alten DT-Wert 90% verwendet, vom neuen DT-Wert 10%. Störungen in den berechneten DT-Werten werden so auf 10% geschrumpft. Nachteil ist hier eine Zeitverzögerung von  $10 \cdot dt$ .

```

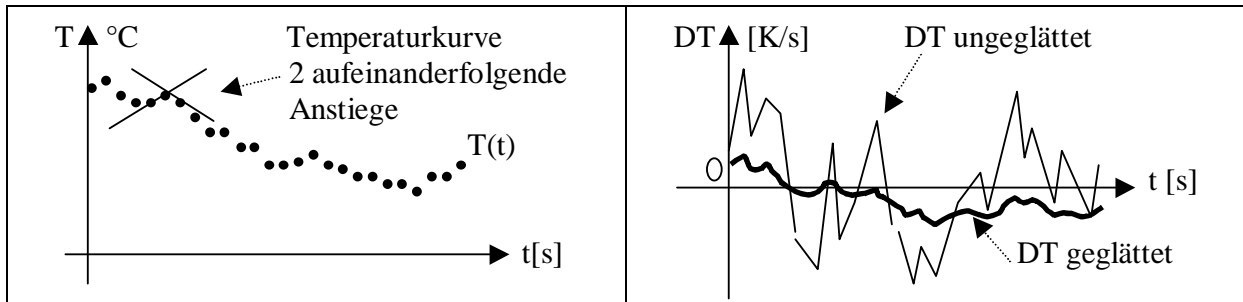
Vor der Steuerschleife:  DT = 0; // Startwert für DT
                        C = 0.9; // Vergessensfaktor der exponentiellen Glättung
                        dt = 0.1; // Tastzeit Sekunden
                        T_alt = Messen(Kanal1); // Temperaturmessung
                        delay(100); // Warte 0.1 Sekunde
  
```

```

In der Steuerschleife:  T = Messen(Kanal1); // Neuen Temperaturwert messen
                        DT = C * DT + (1-C) * (T-T_alt)/dt; // DT berechnen und glätten
                        T_alt = T; // Mache aktuellen Messwert zum alten Messwert
                        delay(100); // Warte 0.1 Sekunde
  
```

Die folgenden beiden Graphiken zeigen eine etwas raue T-Kurve, eine ungeglättete DT-Kurve und eine geglättete DT-Kurve. Ziel der Glättung sollte es sein, dass die DT-Kurve nicht rauer als

die T-Kurve ist. Liegt der Vergessensfaktor C der exponentiellen Glättung sehr nahe bei 1, z.B.  $C=0,999$ , dann wird zwar die Glättung immer besser, aber es tritt ein hoher Zeitverzug von  $dt/(1-C)=1000 \cdot dt=100$  Sekunden zwischen T-Kurve und geglätteter DT-Kurve auf.



### 5.3 Neuronale Netze

Literatur: S. Zakharian, P. Ladewig-Riebler, S. Thoe (1998): Neuronale Netze für Ingenieure, Vieweg. Aus diesem Buch stammt auch das Beispiel zur Regelungstechnik.

Wozu neuronale Netze?

- Zur Approximation von Funktionen ohne explizites mathematisches Modell, z.B. Darstellung von Kennlinien oder von Stabilitätsgrenzen von Reglern
- Für eine Klassifikation von Objekten oder Situationen, z.B. von Regelsituationen
- Zur Optimierung, z.B. einer adaptiven Regleranpassung an sich ständig ändernde Regelstrecken
- In der Bild-, Schrift- und Spracherkennung (Mustererkennung)

Welche Vorteile sieht man z.B. gegenüber Fuzzy-Control?

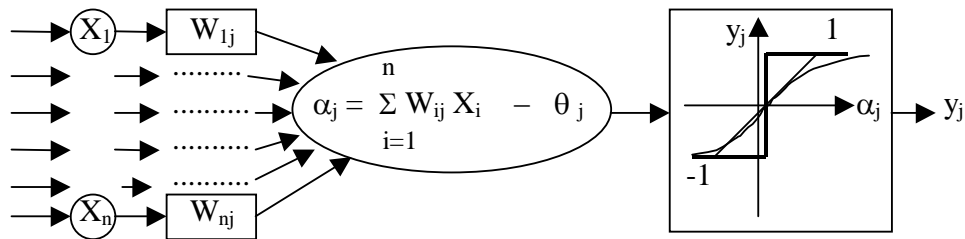
Es gibt nur Standardregeln, d.h., man muss keine Expertenregeln erfragen und notieren. Das neuronale Netz passt in einem Lernvorgang das Netz an die Lerndaten an.

Welche Nachteile gibt es?

Lange Rechenzeiten für das Lernen trotz schneller Prozessoren. Oftmals fehlt beim sogenannten "überwachten Lernen" das Ziel, d.h., es ist in der Regelungstechnik nicht immer klar, was eine *gute Regelung* ist bzw. wer genügend gute Vorlagen liefert..

#### 5.3.1 Das Einzelschicht-Perzeptron nach F. Rosenblatt (1957)

Das Einzelschicht-Perzeptron besteht aus einer Schicht von  $k$  mathematischen Nachbildungen  $y_1, y_2, \dots, y_k$  einer **Nervenzelle**. Das Modell einer einzelnen Nervenzelle  $j$  funktioniert folgendermaßen: Es werden die **Eingangssignale**  $x_1, x_2, \dots, x_n$  des Neurons  $j$  mit den **Gewichten**  $w_{ij}$  gewichtet und addiert. Überschreitet die Gewichtssumme  $\sum w_{ij} x_j$  einen festgelegten **Schwellwert**  $\theta_j$  des Neurons, dann "feuert" das Neuron ein Signal  $y_j$  nach draußen. Die Signalhöhe kann in manchen Modellen mit ganz bestimmten **Aktivierungsfunktionen** von der Gewichtssumme  $\sum w_{ij} x_j$  abhängen. Das folgende Bild zeigt das Modell eines einzelnen Neurons  $y_j$ :



Rechts im Bild sind drei Beispiele von Aktivierungsfunktionen  $y(\alpha)$  eingezeichnet:

1. Die Sprungfunktion, die bei  $\alpha=0$  einen Sprung von  $-1$  nach  $+1$  macht
2. Die Rampe, die in einem Toleranzbereich um  $\alpha=0$  herum linear von  $-1$  nach  $+1$  steigt
3. Eine S-Funktion, die stetig und s-förmig von  $-1$  nach  $+1$  steigt

Das Einzelschicht-Perzeptron besteht aus  $k$  Neuronen  $y_1, y_2, \dots, y_k$ , die alle dieselben Eingangssignale  $x_1, x_2, \dots, x_n$  benutzen, aber schon ein einzelnes Neuron kann z.B. abhängig von der Zahl  $n$  der Eingangssignale verschieden komplizierte Klassifikationsaufgaben lösen, z.B. bei

- $n=1$  eine Zahlengerade durch einen Punkt in zwei Klassen teilen
- $n=2$  eine Ebene durch eine Gerade in zwei Klassen teilen
- $n=3$  einen Raum durch eine Ebene in zwei Klassen teilen

Hat man  $K$  Klassen zu unterscheiden, dann benötigt man mindestens  $K-1$  Neuronen in der Einzelschicht. Überschneiden sich jedoch die Geraden in der Ebene, dann benötigt man sogar mehrere Schichten (siehe Mehrschicht-Perzeptron). Im folgenden Beispiel wird von einem Neuron eine Trenngerade gesucht, die eine Ebene in zwei Klassen teilt. Auf der Ebene sind zwei Arten von Objekten angesiedelt (hier 2 Kreise und 4 Quadrate), die durch die Gerade voneinander abgeteilt werden sollen. Es gibt die zwei Koordinaten  $x_1$  und  $x_2$ . Die folgende kleine Tabelle zeigt die Klasse und die Koordinaten, die Graphik zeigt die Ebene, die Objekte und den zulässigen Bereich für die Trenngerade (schraffiert).

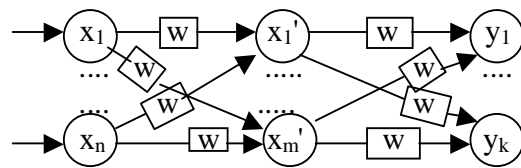
Objekt	Klasse	$x_1$	$x_2$
Kreis	$d=1$	1	5
Kreis	$d=1$	5	4
Quadrat	$d=-1$	-2	0
Quadrat	$d=-1$	-1	0
Quadrat	$d=-1$	5	1
Quadrat	$d=-1$	10	5

Der **Lernalgorithmus** ist eine **Optimierungsaufgabe**, bei der die **Summe der Fehlerquadrate** minimiert wird. Der Algorithmus könnte also auch Lösungen für solche Aufgaben finden, bei denen keine saubere Trennung der Objekte möglich ist. In diesem Falle würde der Algorithmus den **Klassifikationsfehler minimieren**. Lerndatenumfang ist hier  $l=6$ , **Lerndaten** sind unsere 6 Koordinatenpaare und die 6 Klassennummern  $d=1$  bzw.  $d=-1$ . Viele Autoren benutzen noch einen **Lernfaktor**  $\eta$ , der die Konvergenz der Minimumsuche beschleunigen soll. Wenn man nicht weiß, welcher Wert von  $\eta$  optimal ist, nimmt man einfach  $\eta=1$ . Die folgende kleine Tabelle beschreibt den Minimierungsalgorithmus für die Fehlerquadratsumme.

Zeile	Beschreibung	Algorithmus
1	Startgewichte, Schwellenwert, Lernfaktor, j	$w_1 = w_2 = 1, \theta = 2, \eta = 1, j=1$
2	Nimm Zeile j der Lerndaten ( $j=1, 2, \dots, l$ )	$x_{1j} = -2, x_{2j} = 0, d_j = 1$
3	Bilde $\alpha = \sum w_i x_i - \theta$	$\alpha = 1 \cdot (-2) + 1 \cdot 0 - 2 = -4$
4	Bilde y wenn $\alpha < 0$ , dann $y = -1$ wenn $\alpha = 0$ , dann $y = 0$ wenn $\alpha > 0$ , dann $y = +1$	$\alpha = -4 \longrightarrow y(\alpha) = -1$
5	Berechne den Fehler $\Delta = d - y(\alpha)$	$\Delta = 1 - (-1) = 2$
6	Aktualisiere die Gewichte $w_1 \longleftarrow w_1 + \eta \cdot \Delta \cdot x_1$ $w_2 \longleftarrow w_2 + \eta \cdot \Delta \cdot x_2$	$1 + 1 \cdot 2 \cdot (-2) \longrightarrow w_1 = -3$ $1 + 1 \cdot 2 \cdot 0 \longrightarrow w_2 = 1$
7	Wiederhole Zeile 2 bis 6 immer wieder für alle Zeilen der Lerndaten, bis $\sum \Delta_j^2 = \text{Minimum}$ ist	

### 5.3.2 Mehrschicht-Perzeptron

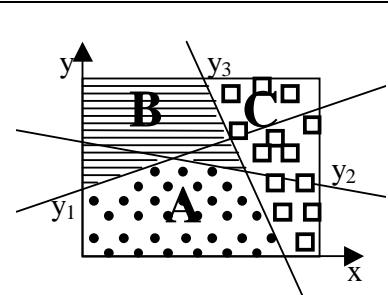
Das Mehrschicht-Perzeptron kann pro Ausgangssignal z.B. eine Gerade bilden (bei  $n=2$  Eingangssignalen  $x_1, x_2$ ) und logische Funktionen, wie AND, OR oder XOR realisieren, d.h. auch komplizierter geformte Klassen erkennen.



Die Zahl der Neuronen in jeder Schicht kann unterschiedlich sein. Dabei wird jedes Neuron der verborgenen ersten Schicht (hidden layer)  $x_1', x_2', \dots, x_m'$  über Gewichte  $w_{ij}$  mit allen  $n$  Eingangssignalen  $x_1, \dots, x_n$  und über andere Gewichte  $w_{jq}$  mit allen  $k$  Neuronen  $y_1, \dots, y_k$  der zweiten Schicht verbunden, so dass man insgesamt  $n \cdot m + m \cdot k$  verschiedene Gewichte  $w$  benötigt. Deren Berechnung aus zufälligen Startgewichten ist eine anspruchsvolle Optimierungsaufgabe, die

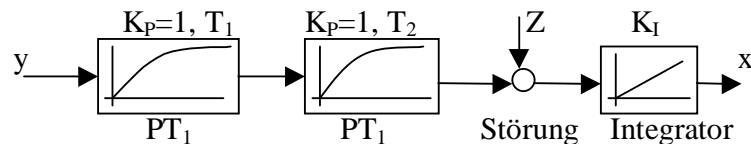
1. nicht immer das absolute Minimum findet
2. oft sehr viel Rechenzeit in Anspruch nimmt

Die Graphik rechts zeigt drei unterschiedlich geformte Objektklassen A, B, C, die durch 3 Geradengleichungen der Form  $g_j = w_{1j} \cdot x_1 + w_{2j} \cdot x_2$  abgegrenzt werden.  $g_1 = w_{11} \cdot x_1 + w_{21} \cdot x_2$  lässt sich umformen in die übliche Geradengleichung  $x_2 = B_{10} + B_{11} \cdot x_1$  bzw. mit  $y = x_2, x = x_1$  in  $y_1 = B_{10} + B_{11} \cdot x$ . Ebenso erhalten wir  $y_2 = B_{20} + B_{21} \cdot x$  und  $y_3 = B_{30} + B_{31} \cdot x$ . Objekte von A finden wir mit  $y < y_1$  UND  $y < y_2$  UND  $y < y_3$ . Objekte von B finden wir mit  $y \geq y_1$  UND  $y \geq y_2$  UND  $y < y_3$ . Objekte von C finden wir mit  $y \geq y_3$ .



### 5.3.3 Ein Beispiel aus der Regelungstechnik

Ein Roboterarm hebt 4 Sorten unterschiedlich schwerer Teile aus einem Behälter auf ein Band, und das möglichst ohne Überschwingen wegen der Stauch- und Kollisionsgefahr. Die Regelstrecke *Roboterarm* ist in diesem Beispiel hier eine  $IT_2$ -Strecke (siehe Skript Regelungstechnik zur Abkürzung  $IT_2$ ) mit dem Blockschaltbild



Die  $IT_2$ -Strecke wurde für die 4 unterschiedlich schweren Teile P1, P2, P3, P4 jeweils identifiziert, d.h. die Streckenparameter  $T_1$ ,  $T_2$  und  $K_I$  bestimmt. Da die Integrierkonstante  $K_I$  nur vom Übersetzungsverhältnis des Motorgetriebes des Roboterarms abhängt, und somit konstant ist, ergaben sich lediglich unterschiedliche Verzögerungszeiten  $T_1$  und  $T_2$ , wie die nachfolgende kleine Tabelle zeigt.

	P1	P2	P3	P4
$T_1$	0,10	0,220	0,700	1,44
$T_2$	0,19	0,135	0,105	0,10

I-Strecken regelt man mit PD-Reglern, wenn das Messsignal rauschfrei verfügbar ist. Das Messsignal ist hier die Rückmeldung über die Position des Teils. Für jedes unterschiedliche Teil müsste man eigentlich mit einem speziell auf dieses Teil abgestimmten PD-Regler arbeiten. Die optimalen Parameter (optimale Reglerverstärkung  $K_{PR}$  und optimale Vorhaltezeit  $T_V$  des D-Anteils) für die 4 Regelsituationen zeigt die folgende kleine Tabelle:

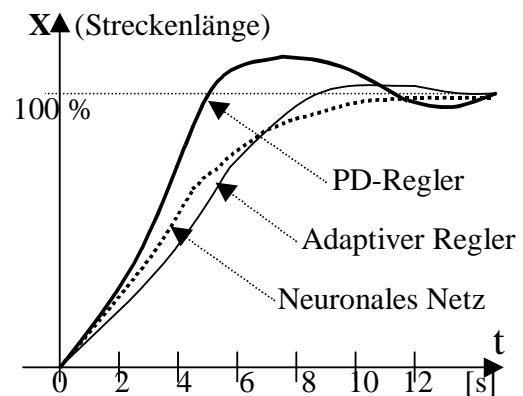
	P1	P2	P3	P4
$K_{PR}$	5,26	7,41	9,52	10,0
$T_V$	0,19	0,135	0,105	0,1

Das Problem bei dieser Aufgabe ist, dass der Regler nicht den Typ des Teils weiß, das am Robotergreifer hängt. Ist das Teil leicht, schwingt der Arm über und knallt das Teil aufs Band. Ist das Teil schwer, schafft der Roboter die Zeitvorgabe nicht.

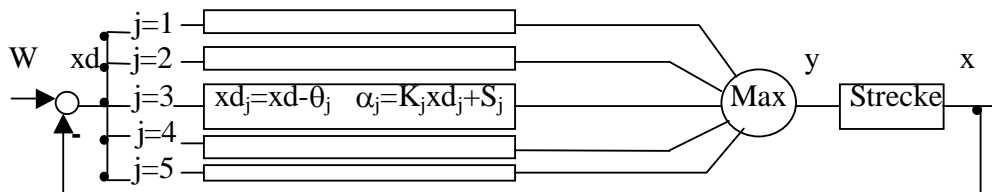
Die folgende kleine Graphik zeigt das Ergebnis von drei unterschiedlichen Lösungen:

1. Kompromiss beim PD-Regler, d.h. ein gemittelttes  $K_{PR}$  und gemittelttes  $T_V$
2. Ein adaptiver PD-Regler, der aus der Anfangsdynamik  $x(y(t))$  die Reglerparameter schätzt
3. Eine Neuronale-Netz-Lösung, die mit der dargestellten Einlaufkurve als Zielvorgabe "angelernt" wurde.

1. Für einen einfachen PD-Regler muss man einen Kompromiss eingehen. Die Weg-Zeitkurve des PD-Reglerkompromisses für ein leichtes Teil zeigt dann auch ein deutliches Überschwingen.
2. Ein adaptiver Regler berechnet aus der Weg-Zeit-Kurve der ersten 0,1 Sekunden  $T_1$  und  $T_2$  und daraus ein angepasstes  $K_{PR}$  und  $T_V$ . Aber auch er schwingt noch leicht über.
3. Der von einem neuronalen Netz gesteuerte Roboterarm erkennt die Bewegungsdynamik und reagiert wie ein geschulter Kranfahrer



Das Blockschaltbild der Neuronalen-Netz-Steuerung zeigt die folgende Graphik:



Sollwert  $W$  ist der Zielpunkt auf dem Band, d.h. ein ganz bestimmter Wert von  $x$ . Die Regeldifferenz  $xd = W - x$  wird in 5 Signale aufgesplittet und es werden parallel 5 Alpha-Signale  $\alpha_1, \alpha_2, \dots, \alpha_5$  berechnet. Dafür werden 2-5 Gewichte  $K_j$  und  $S_j$  und 5 Schwellenwerte  $\theta_j$  benötigt, die in einer Lernphase vor dem ersten Einsatz des Roboters bestimmt werden müssen. Die Zahl 5 ist eher pragmatisch zu sehen. Es hätten auch 6 oder 4 sein können. Die Maximumbildung ist eine Anleihe aus der Fuzzy Logic und entspricht einer ODER-Verknüpfung der 5 Stellsignale  $\alpha_j$ .

## 5.4 Mehrgrößenregelung

Für jede Regelgröße in einer Anlage, z.B. in einem Fermenter, muss eine eigene Stellgröße existieren. Die Temperatur regelt man mit Heizung/Kühlung, die  $O_2$ -Konzentration mit der Rührerdrehzahl, den pH-Wert mit der Drehzahl der Dosierpumpe, den Druck mit dem Hub des Ablassventils usw. Auf diese Weise existieren in einer Anlage mehrere Regelkreise, die im Normalfall nebeneinander arbeiten, ohne sich gegenseitig zu stören. Hier spricht man von Einzelregelkreisen,

Sind jedoch starke Wechselwirkungen (Kopplungen) zwischen den einzelnen Regelkreisen vorhanden, dann ist *Mehrgrößenregelung* angesagt. Die Mehrgrößenregelung befasst sich mit der Regelung stark gekoppelter Regelkreise.

Beispiel: Heizen erniedrigt die  $O_2$ -Konzentration im Fermenter, da warmes Wasser weniger Sauerstoff lösen kann, als kaltes. Erhöht man jetzt die Rührerdrehzahl, wird zwar mehr Sauerstoff eingetragen, aber auch zusätzliche Reibungswärme. Etwa 90% der Rührermotorleistung bleiben zuerst einmal als Wärme im Fermenter. (Ob die Kopplung in diesem Beispiel wirklich so stark ist, dass man Mehrgrößenregelung anwenden muss, wird im Einzelfall entschieden.)

Ein Koppelmaß  $K_{ij}$  gibt an, wie stark eine Änderung der Regelgröße  $X_j$  die Regelgröße  $X_i$  beeinflusst. Dabei gilt, dass allgemein  $K_{ij} \neq K_{ji}$  ist.

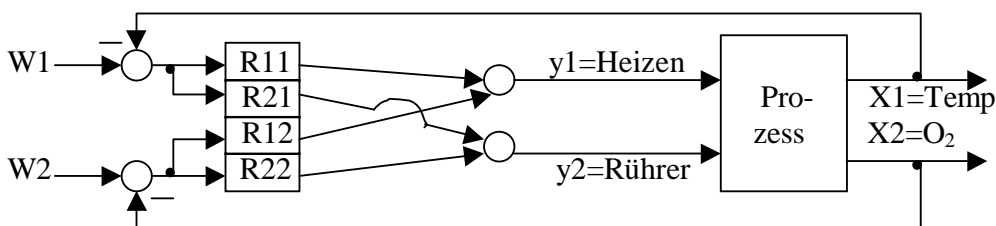
$K_{ij} = \frac{\Delta X_i}{\Delta X_j} \frac{D_j}{D_i}$	
--	--

$\Delta X_i$  und  $\Delta X_j$  sind Änderungen der Regelgrößen  $X_i$  und  $X_j$  nach einem aufgegebenen  $y_j$ -Sprung. Eigentlich sollte durch diesen Stellgrößenprung  $U_{0j} = \Delta y_j$  nur die Regelgröße  $X_j$  beeinflusst werden, aber durch die Kopplung ändert sich Regelgröße  $X_i$  ungewollt mit.  $D_i$  und  $D_j$  sind die Toleranzen für die Regelgrößen  $X_i$  und  $X_j$ . Man nimmt die gesamte Breite des Toleranzbandes. Z.B. sei bei einer Temperaturregelung die zulässige Toleranz  $\pm 2^\circ\text{C}$ , dann ist  $D = 4^\circ\text{C}$ . Ab einem Koppelmaß  $K_{ij} > 0.5$  spricht man von starker Kopplung und sollte eine Mehrgrößenregelung in Erwägung ziehen.

Es gibt zahlreiche Verfahren der Mehrgrößenregelung, z.B.

- Stationäre Entkopplung
- Dynamische Entkopplung
- Fuzzy Control
- Beobachtermodelle usw.

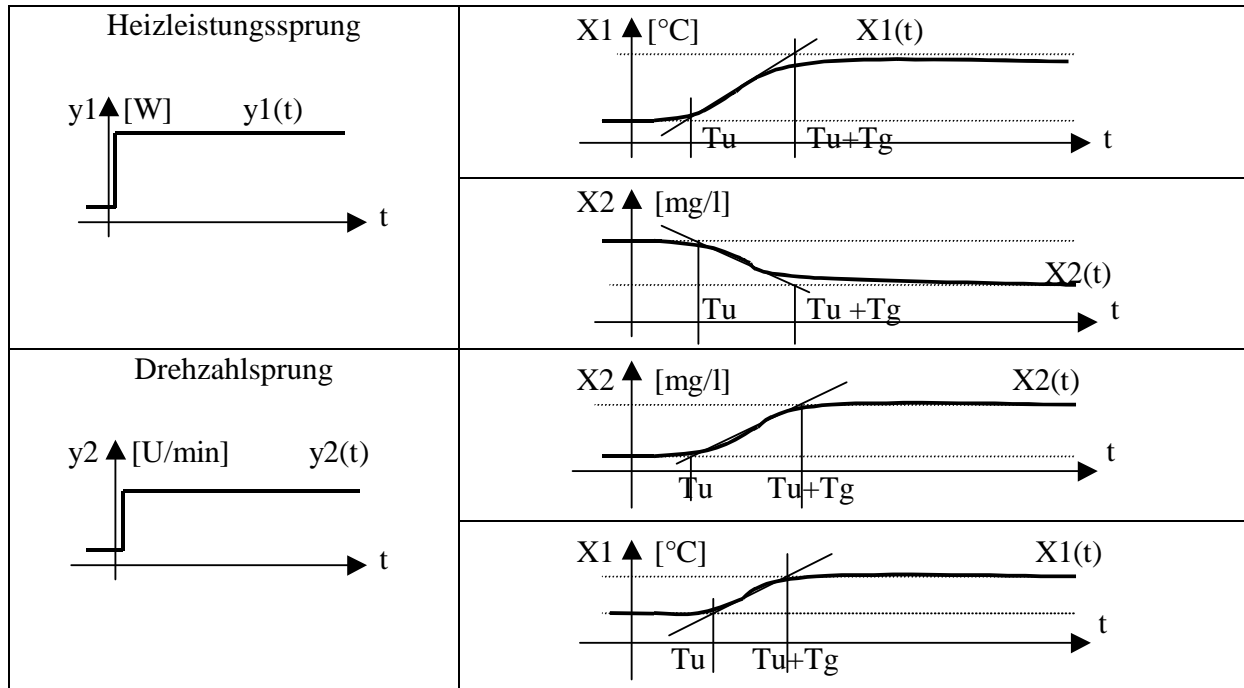
Die *stationäre Entkopplung* benutzt für stark gekoppelte Regelkreise zusätzliche Entkopplungsregler. Wie der Name sagt, funktioniert die Entkopplung am besten im stationären Zustand (wenig Störungen, wenig Sollwertänderungen). Bei Sollwertänderungen oder starken Störungen können kurzzeitig Schwingungen oder ein Überschwingen auftreten. Ein gutes Beispiel ist die stationäre Entkopplung nach Boksenbom und Hood. Die nachfolgende Graphik zeigt zwei Einzelregelkreise für Temperatur und  $\text{O}_2$ -Konzentration, die durch zwei Zusatzregler entkoppelt werden.



- $W_1, W_2$  sind die Sollwerte für Temperatur und  $\text{O}_2$ -Konzentration  
 $X_1, X_2$  sind die Istwerte für Temperatur und  $\text{O}_2$ -Konzentration  
 $y_1$  ist die Stellgröße für die Heizung (Heizleistung)  
 $y_2$  ist die Stellgröße für die  $\text{O}_2$ -Konzentration (Rührerdrehzahl)  
 $R_{11}, R_{22}$  sind die Hauptregler des Temperatur- und des  $\text{O}_2$ -Regelkreises  
 $R_{12}, R_{21}$  sind die beiden Entkopplungsregler

Die Berechnung der Haupt- und Entkopplungsregler erfolgt in 4 Schritten:

**Schritt 1:** Ausführung von zwei Sprungversuchen und Registrierung der 4 Sprungantworten und daraus die Bestimmung der  $\mathbf{K}_{PS}$ -Matrix der Streckenverstärkungen.



Die  $\mathbf{K}_{PS}$ -Matrix wird aus den 4 Verstärkungen der 4 Sprungantworten gebildet und sei

$$\mathbf{K}_{PS} = \begin{bmatrix} 0,09 \left[ \frac{K}{W} \right] & 0,008 \left[ \frac{K}{U / \min} \right] \\ -0,0045 \left[ \frac{mg / l}{W} \right] & 0,170 \left[ \frac{mg / l}{U / \min} \right] \end{bmatrix}.$$

**Schritt 2:** Auslegung von P-Reglern für die beiden Einzelregelkreise so, als wäre keine Kopplung vorhanden (z.B. nach der Reglerparametrierung von Chien, Hrones und Reswick aus dem RT-Skript). Die beiden berechneten Verstärkungen  $K_{P11}$  und  $K_{P22}$  werden als Diagonalmatrix  $\mathbf{D}_R$  geschrieben. Diese sei

$$\mathbf{D}_R = \begin{bmatrix} 1,7 \left[ \frac{W}{K} \right] & 0 \\ 0 & 4,1 \left[ \frac{U / \min}{mg / l} \right] \end{bmatrix}$$

**Schritt 3:** Berechne die Reglermatrix nach der Matrix-Formel  $\mathbf{R} = \mathbf{K}_{PS}^{-1} \mathbf{D}_R \mathbf{D}_K$ .

Hierbei ist  $\mathbf{K}_{PS}^{-1}$  die inverse Matrix zu  $\mathbf{K}_{PS}$ . Man erhält sie mit EXCEL oder einem geeigneten Taschenrechner.  $\mathbf{D}_K$  ist die Diagonale von  $\mathbf{K}_{PS}$  als Diagonalmatrix geschrieben. Die numerische Berechnung erfordert zwei Matrixmultiplikationen und ist

$$\mathbf{R} = \begin{pmatrix} 11,085 & -0,5216 \\ 0,2934 & 5,8685 \end{pmatrix} \cdot \begin{pmatrix} 1,7 & 0 \\ 0 & 4,1 \end{pmatrix} \cdot \begin{pmatrix} 0,09 & 0 \\ 0 & 0,17 \end{pmatrix}$$

bzw.

$$R = \begin{pmatrix} 1,70 \left[ \frac{W}{K} \right] & -0,36 \left[ \frac{W}{mg/l} \right] \\ 0,04 \left[ \frac{U/min}{K} \right] & 4,1 \left[ \frac{U/min}{mg/l} \right] \end{pmatrix}$$

Wie man sieht, haben sich die Hauptregler R11 und R22 nicht verändert, aber es wurden zwei Entkopplungsregler berechnet, von denen R12 eine negative Verstärkung aufweist, d.h. der Ausgang dieses Reglers muss subtrahiert werden, nicht addiert.

Schritt 4: Ersetze die 4 P-Regler R11=1.70, R12= -0.36, R21=0.04 und R22=4.1 durch PI- bzw. PID-Regler. Ein Beispiel soll die Vorgehensweise demonstrieren. Die Reglertabelle von Chien, Hrones und Reswick (siehe RT-Skript) hat die zwei Spalten "Führungs-" bzw. "Konstantregelung", und diese wieder unterteilt in je zwei Spalten "0% Überschwingen" und "20% Überschwingen". Wir benutzen die Spalte "Führungsregelung mit 0% Überschwingen".

Laut Tabelle gilt für P-Regler  $V_0 T_u / T_g = 0,3$

Laut Tabelle gilt für PI-Regler  $V_0 T_u / T_g = 0,6$  und  $T_n = 4 T_u$

Für den Hauptregler R11 = 1.70 [W/K] sei aus der y1-Sprungantwort der Temperatur (siehe Graphik oben) eine Verzugszeit  $T_{u11} = 7$  [s] bestimmt worden. Wir berechnen

$$K_{p11} = \frac{R11}{0,3} \cdot 0,6 = \frac{1,70}{0,3} \cdot 0,6 = 3,40 \left[ \frac{W}{K} \right] \quad \text{Reglerverstärkung}$$

$$T_{n11} = 4 \cdot T_{u11} = 28 \text{ [s]} \quad \text{Nachstellzeit des I-Anteils}$$

Für einen der Koppelregler, z.B. R12=-0.36 [W/(mg/l)], machen wir eine ähnliche Rechnung. Hier wird die Verzugszeit aus der Temperatur-Sprungantwort zum Rührerdrehzahlprung y2(t) abgelesen (siehe Graphik oben) und sei  $T_{u12} = 9$  [s].

$$K_{p12} = \frac{R12}{0,3} \cdot 0,6 = \frac{-0,36}{0,3} \cdot 0,6 = -0,72 \left[ \frac{W}{mg/l} \right] \quad \text{Reglerverstärkung}$$

$$T_{n12} = 4 \cdot T_{u12} = 36 \text{ [s]} \quad \text{Nachstellzeit des I-Anteils}$$

Ist ein Koppelkoeffizient  $K_{ij} < 0.3$ , dann kann auf den Entkopplungsregler  $R_{ij}$  verzichtet werden. Bei einer Softwarelösung ist die Einsparung unerheblich. Werden jedoch Hardware-Regler benutzt, dann lohnt sich diese Überlegung.

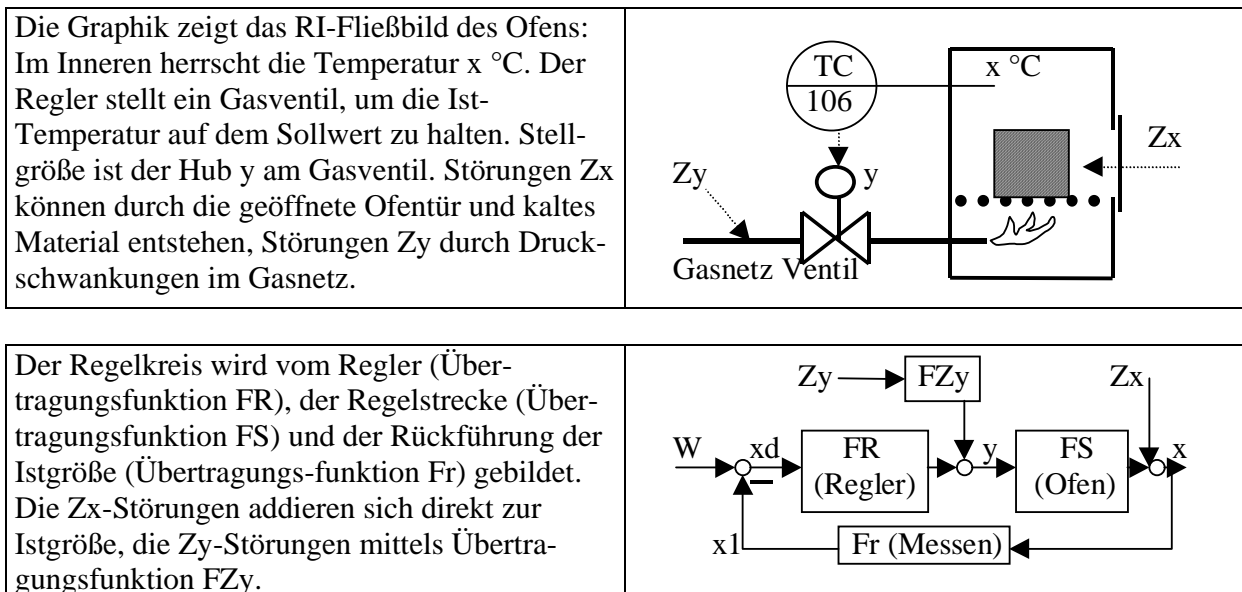
## 5.5 Stabilität und Güte von Regelkreisen

Stabilität in der Regelung heißt, dass keine Dauerschwingungen auftreten, und auch kein Aufschaukeln von Schwingungen möglich ist.

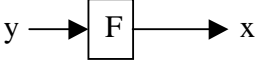
Güte einer Regelung heißt Einhalten des Toleranzbandes für den Istwert, und ein schneller Übergang ohne Schwingungen bei Sollwertänderungen.

Stabilität und Güte widersprechen sich teilweise, zumindest was Geschwindigkeit und Schwingungsfreiheit anbetrifft. Der Ingenieur muss hier einen Kompromiss suchen.

### 5.5.1 Beispiel Temperaturregelung eines Industrieofens



W	Sollwert, z.B. 1000 °C Schmiedetemperatur für Stahlrohlinge
x	tatsächliche Temperatur im Ofen
x1	Istwert nach dem Messvorgang, z.B. mit einem Thermoelement NiCrNi gemessen
$x_d = W - x_1$	Regeldifferenz Sollwert minus Istwert
y	Stellgröße, z.B. der Hub des Gasventils in % von 100
$Z_x$	Störgröße, z.B. die Temperaturabsenkung in °C durch eine offene Ofentür oder einen kalten Stahlblock
$Z_y$	Störgröße, z.B. eine Druckminderung im Gasnetz. Z.B. könnten sich 10% Minderdruck wie 15% weniger Hub am Gasventil auswirken. Deshalb rechnet die Übertragungsfunktion FZy Druckschwankungen [mbar] in y-Werte [% Hub] um.
FR	Übertragungsfunktion des Reglers, z.B. eines PID-Reglers
FS	Übertragungsfunktion der Regelstrecke einschließlich Stellantrieb (hier der Ofen und das Regelventil), modelliert als PT <sub>2</sub> -Strecke
Fr	Übertragungsfunktion der Rückführung, z.B. die Verzögerung der Temperaturmessung durch die Keramikummantelung des Thermoelements modelliert als PT <sub>1</sub> -Strecke

Zur Erinnerung: Die Übertragungsfunktion eines Gliedes ist der Quotient von Ausgangssignal $x$ und Eingangssignal $y$ .	$F(p) = \frac{x(p)}{y(p)}$	
---	----------------------------	---

An dieser Stelle soll aus dem Blockschaltbild die **Kreisgleichung** abgeleitet werden, die im RT-Skript ohne Ableitung einfach angegeben wurde. Liegen Übertragungsglieder in Reihe, dann multiplizieren sich die Übertragungsfunktionen (siehe RT-Skript). Wir arbeiten im Frequenzbereich, d.h. wir schreiben  $x(p)$  statt  $x(t)$ ,  $W(p)$  statt  $W(t)$  usw. Laut obigem Blockschaltbild ist

$$x(p) = Zx(p) + FZy(p) \cdot FS(p) \cdot Zy(p) + FR(p) \cdot FS(p) \cdot xd(p) .$$

Das Signal  $Zy$  muss erst durch die Box  $FZy$  und dann durch die Box  $FS$  hindurch, bevor es zu einem Bestandteil von  $x$  wird. Das Signal  $xd$  muss erst durch die Box  $FR$  und dann durch die Box  $FS$  hindurch, bevor es zu einem Bestandteil von  $x$  wird. Lediglich die Störung  $Zx$  ist direkter Bestandteil von  $x$ . Wir ersetzen  $xd(p)$  durch  $W(p) - x_1(p)$  und  $x_1(p)$  durch  $Fr(p) \cdot x(p)$ , da die reale Temperatur  $x$  durch die Box  $Fr$  hindurch muss, um zur gemessenen Istgröße  $x_1$  zu werden, und wir erhalten die Gleichung

$$x(p) = Zx(p) + FZy(p) \cdot FS(p) \cdot Zy(p) + FR(p) \cdot FS(p) \cdot [ W(p) - Fr(p) \cdot x(p) ] .$$

Wir lösen die eckige Klammer auf, bringen den Term  $FR(p) \cdot FS(p) \cdot Fr(p) \cdot x(p)$  zum  $x(p)$  auf die linke Seite und klammern dort  $x(p)$  aus, und erhalten so

$$x(p) \cdot [ 1 + FR(p) \cdot FS(p) \cdot Fr(p) ] = Zx(p) + FZy(p) \cdot FS(p) \cdot Zy(p) + FR(p) \cdot FS(p) \cdot W(p)$$

Nun dividieren wir beide Seiten der Gleichung durch die Klammer  $[ 1 + FR(p) \cdot FS(p) \cdot Fr(p) ]$  und erhalten die Kreisgleichung für unseren gestörten Regelkreis

$$x(p) = \frac{FR(p)FS(p)}{1 + FR(p)FS(p)Fr(p)} W(p) + \frac{FZy(p)FS(p)}{1 + FR(p)FS(p)Fr(p)} Zy(p) + \frac{Zx(p)}{1 + FR(p)FS(p)Fr(p)}$$

Die 3 Terme der rechten Seite beschreiben die Einflüsse des Sollwertverlaufs und die Einflüsse von  $Zy$ - und  $Zx$ -Störungen auf die Temperatur  $x$  im Ofen.

### 5.5.2 Zahlenbeispiel für den Einfluss von Sollwertänderungen

PID-Regler mit  $K_{PR} = 10,0$  [% / K] Reglerverstärkung in Hub pro Grad Abweichung  
 $T_N = 17,5$  [s] Nachstellzeit des I-Anteils des PID-Reglers  
 $T_V = 3,5$  [s] Vorhaltezeit des D-Anteils des PID-Reglers  
 und der Übertragungsfunktion

$$FR(p) = K_{PR} \left[ 1 + \frac{1}{T_N p} + T_V p \right] = 10,0 \left[ 1 + \frac{1}{17,5 p} + 3,5 p \right]$$

Ofen mit  $PT_2$ -Verhalten  $K_{PS} = 16,2$  [K / %] Streckenverstärkung in Grad pro Ventilhub  
 $T = 8,5$  [s] Zeitkonstante eines Verzögerungsgliedes

und der Übertragungsfunktion

$$FS(p) = K_{PS} \left( \frac{1}{1 + pT} \right)^n = 16,2 \left( \frac{1}{1 + 8,2 \cdot p} \right)^2$$

Messung mit  $PT_1$ -Verhalten  $K_p = 1,0$  [K/K] Verstärkungsfaktor  
 $T = 2,7$  [s] Zeitkonstante der Messverzögerung  
 und der Übertragungsfunktion

$$Fr(p) = \frac{K_p}{1 + Tp} = \frac{1}{1 + 2,7p}$$

Damit ist die Übertragungsfunktion  $F(p)$  des Kreises bezüglich einer Sollwertänderung auch numerisch berechenbar:

$$x(p) = \frac{FR(p)FS(p)}{1 + FR(p)FS(p)Fr(p)} W(p) = F(p)W(p) .$$

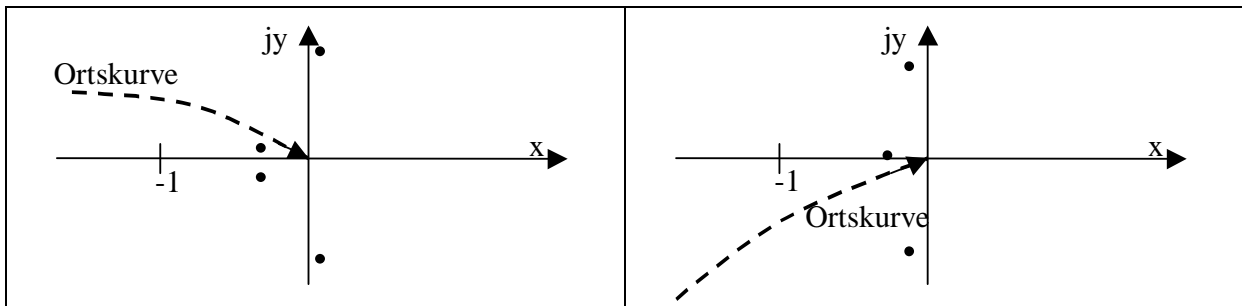
### 5.5.3 Zwei gleichwertige Stabilitätskriterien

Für die Beurteilung der Stabilität eines Regelkreises ist vor allen Dingen der Nenner der Übertragungsfunktion,  $1 + FR(p) \cdot FS(p) \cdot Fr(p)$ , wichtig. Das Produkt  $F_0(p) = FR(p) \cdot FS(p) \cdot Fr(p)$  heißt *Übertragungsfunktion des aufgeschnittenen Regelkreises* und spielt eine wichtige Rolle.  $F_0(p)$  beschreibt die Reihenschaltung von Regler, Regelstrecke und Rückführung (siehe Blockschaltbild oben).

Das **Wurzelort-Kriterium** der Stabilität (Root Locus Criterion): Sind die Realteile aller Wurzeln (Nullstellen) der Gleichung  $1 + F_0(p)$  negativ, dann treten nur gedämpfte Schwingungen auf, und die Regelung ist stabil.

Das **Ortskurven-Kriterium** (auch Nyquist- oder Routh-Hurwitz Criterion) : Lässt die Ortskurve  $F_0(j\omega)$  für  $\omega \rightarrow \infty$  den Punkt "-1" auf der reellen Achse "links liegen", dann ist keine Selbstverstärkung möglich und die Regelung ist stabil.

<p>Eine Lösung mit den PID-Parametern <math>K_{PR}=30</math>, <math>T_N=8</math> und <math>T_V=1,5</math> zeigt zwei konjugiert komplexe Wurzeln im positiven Bereich bei <math>0,03 \pm 1,93j</math> und zwei konjugiert komplexe Wurzeln im negativen Bereich bei <math>-0,33 \pm 0,15j</math>. Die Ortskurve lässt den Punkt "-1" rechts liegen. Beide Kriterien sagen <b>instabil</b>.</p>	<p>Eine Lösung mit den PID-Parametern <math>K_{PR}=10</math>, <math>T_N=17,5</math> und <math>T_V=3,5</math> zeigt zwei konjugiert komplexe Wurzeln im negativen Bereich bei <math>0,16 \pm 1,70j</math> und eine reelle Wurzel im negativen Bereich bei <math>-0,21</math>. Die Ortskurve lässt den Punkt "-1" links liegen. Beide Kriterien sagen <b>stabil</b>.</p>
--	--



**Wie berechnet man die Wurzeln (Nullstellen)** einer komplexen Funktion  $F(p)$ ? Systeme wie Mathcad oder Maple bieten da Lösungen an. Wer es selbst einmal versuchen will, geht z.B. folgendermaßen vor. Man nimmt den Bildschirm als Gaußsche komplexe Zahlenebene und steckt den Zahlenbereich für das komplexe Argument  $p = \delta + \omega j$  ab, z.B.  $-3 \leq \delta \leq 3$  und  $-3 \leq \omega \leq 3$ . Auf diese Weise kann man jedem Pixel einen komplexen Wert zuordnen. Den setzt man in die komplexe Funktion  $F(p)$  ein, berechnet deren Wert  $z = x + y j$ . Dann bildet man den Betrag

$$|F(p)| = |z| = \sqrt{x^2 + y^2}.$$

Den Betrag kann man jetzt zur Färbung in einer Grauskala desjenigen Pixels benutzen, zu dem das  $p$  gehört. Kleine Beträge färbt man dunkel, denn das sind die Pixel auf oder direkt neben einer Nullstelle. Je größer der Betrag, desto heller färbt man die Pixel.

**Wie berechnet man die Ortskurve** einer Übertragungsfunktion? Die Ortskurve einer komplexen Funktion  $F(p)$  entsteht, wenn man nacheinander  $p$ -Werte in die Funktion einsetzt, die auf der imaginären Achse der komplexen Zahlenebene liegen, d.h.  $p = 0 + \omega j$ . Lässt man z.B.  $\omega$  Werte von 0,0001 bis 10,0 in Schritten von 0,0001 durchlaufen, entstehen 100.000 komplexe  $F(p)$ -Werte  $z = x + y j$ . Die Kurve wird sichtbar, wenn man die Pixel, auf die  $z$  zeigt, schwarz färbt.

#### 5.5.4 Gütemaße der Regelung, Optimierung

Einziges Gütemaß im eingeschwungenen Zustand ist die **Regelabweichung**  $x_d = W - x$ . Diese tritt aber nur bei Reglern ohne I-Anteil auf, z.B. bei einem P-Regler.

Gütemaße für den Einschwingvorgang gibt es reichlich. Schnellen Reglern gestattet man, dass sie bis zu 20% überschwingen, bezogen auf die Höhe eines Sollwertsprungs. Regler ohne Überschwingen arbeiten langsamer.

Bei schnellen Reglern minimiert man eines der folgenden Gütemaße  $T_{an}$ ,  $I$  oder  $I^2$ :

$T_{an}$  = Anregelzeit (Zeit bis das neue Toleranzband  $W_{neu} \pm Tol$  erreicht ist)

$I = \int_0^{TM} |x_d(t)| dt$  (Regelfläche absolut innerhalb einer festgelegten Messzeit TM)

$I^2 = \int_0^{TM} x_d^2(t) dt$  (Quadratische Regelfläche innerhalb einer festgelegten Messzeit TM)

unter Einhaltung einer der folgenden Bedingungen

$\delta \geq 0,45$  (Dimensionslose Dämpfung,  $\delta = 0,45$  gibt gerade 20% Überschwingen)

$X_m \leq 20\%$  (Überschwingweite bezogen auf die Höhe des Sollwertsprungs)

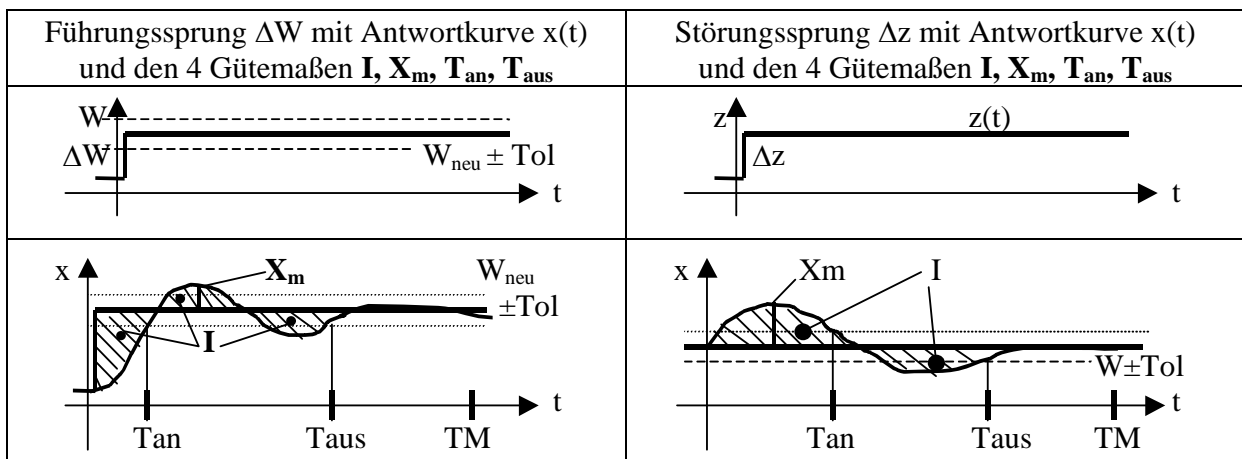
Bei langsamen Reglern ohne Überschwingen minimiert man eines der folgenden Gütemaße:

$T_{\text{aus}}$  = Ausregelzeit (ab  $T_{\text{aus}}$  liegt der Istwert nur noch im Toleranzband  $W_{\text{neu}} \pm \text{Tol}$  um den neuen Sollwert)

$It = \int_0^{TM} t \cdot |xd(t)| dt$  (Zeitgewichtete absolute Regelfläche innerhalb einer festgelegten Messzeit  $TM$ . Spätere Abweichungen werden hier härter bestraft.)

unter Einhaltung von  $X_m \leq \text{Tol}$  (Kein Überschwingen über das Toleranzband hinaus).

Die folgenden 4 kleinen Graphiken zeigen einen Sollwertsprung  $\Delta W$  (Führungssprung) mit der Antwortkurve  $x(t)$  des Regelkreises und eingezeichneten Gütemaßen, sowie einen Störungssprung  $\Delta z$  (z.B. in der Sommerhitze ein Laborfenster aufgerissen) mit der Antwortkurve und eingezeichneten Gütemaßen.



## Optimierung

Optimierung heißt entweder Verbesserung der Regeleigenschaften oder Erhöhung der Stabilität.

Man optimiert **am laufenden Regelkreis** durch vorsichtige Veränderung der Regelparameter und Messung eines der obigen Gütemaße. Beim PID-Regler und zu starkem Überschwingen verringert man die Verstärkung  $K_{PR}$  und vergrößert parallel dazu die Nachstellzeit  $T_N$ . Beim PD-Regler und zu starkem Überschwingen verringert man nur die Verstärkung  $K_{PR}$ . Schleicht der Regler sich zu langsam an den neuen Sollwert heran oder braucht er zu lange, um eine Störung wegzuregulieren, dann erhöht man  $K_{PR}$  und verringert parallel dazu die Nachstellzeit  $T_N$ .

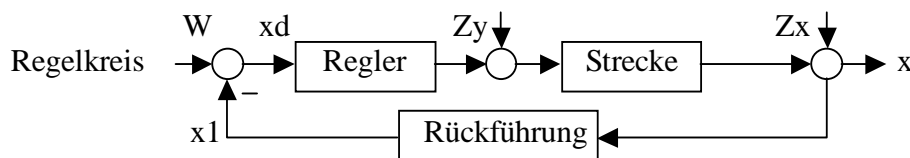
PID-Regler nimmt man für P- und PTn-Strecken bei glattem Messwertverlauf. PI-Regler nimmt man statt PID, wenn das Messsignal gestört ist. In diesem Falle darf nicht differenziert werden.

PD-Regler nimmt man für I- und ITn-Strecken bei glattem Messwertverlauf. Den reinen P-Regler für diese Strecken bei gestörten Messwerten. Bei der Vorhaltezeit  $T_v$  sollte man sich ungefähr am Richtwert  $0,25 \Sigma Ti$  orientieren, wobei  $\Sigma Ti$  die Zeitkonstantensumme der Regelstrecke ITn sind. P- und PD-Regler sollte man immer im Arbeitspunkt betreiben. Das ist der Wert  $y_0$  der Stellgrö-

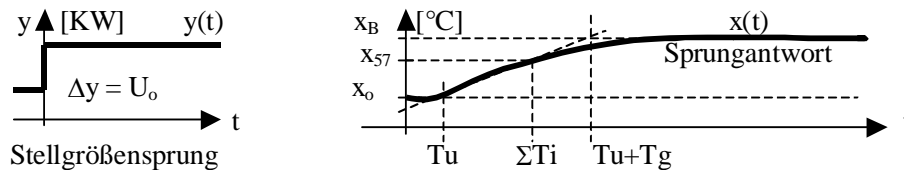
ße, bei dem der Prozess auch ohne Regler liefere, vorausgesetzt, dass keine Störungen und keine Sollwertänderungen anliegen.

Man kann einen Regelkreis auch **am PC optimieren**, wenn man alle Glieder des Regelkreises durch ein Modell beschreiben kann, z.B. die Regelstrecke einschließlich des Aktors durch ein  $PT_2$ -Glieder, die Rückführung (Sensor, Messumformer) durch ein  $PT_1$ -Glieder. Natürlich müssen die echten Glieder dafür identifiziert werden (siehe Skript RT). Alle Verstärkungen und Zeitkonstanten müssen bekannt sein. Man nimmt dann ein Simulationspaket für Regelschaltungen, z.B. IAS oder LabView, programmiert das Regelkreismodell durch Aneinanderklicken der Bausteine, parametrisiert sie mit den identifizierten Verstärkungen und Zeitkonstanten, und kann dann in aller Ruhe den geeigneten Regler ausprobieren und die Antwortkurven des Regelkreises optimieren, indem man die Reglerparameter geeignet wählt.

### Beispiel Temperaturregelung



Um Typ und Parameter der Strecke zu bestimmen, macht man mit der vom Regelkreis abgekoppelten Regelstrecke einen Sprungversuch:



## 6. Bildverarbeitung in der Automatisierungstechnik

### 6.1 Strichcodes (Barcodes)

Etwa seit 1980 zur Identifikation von Waren benutzt.

1-D-Codes: 1-dimensional, meist Striche und Lücken nebeneinander, z.B. auf Etiketten

2-D-Codes: 2-dimensional wie ein Schachbrett oder ein Kreuzworträtsel, z.B. Semacode

Höherdimensionale Codes benutzen zusätzlich Farbe, Helligkeit, Zeit (Animierte Codes)

**EAN-Code** (European Article Number) gibt es als EAN-8 mit 8 Ziffern und als EAN-13.

Weitere Codes sind z.B. Code-39 (Automobilbranche, Pharmaindustrie) oder Code-128, der mittels Umschaltzeichen alle 128 Zeichen des ASCII-Codes darstellen kann.

**Ein Barcodescanner** arbeitet entweder mit einem Laserstrahl, der durch einen rotierenden Polygonspiegel (Spiegel mit vielen Flächen) immer wieder abgelenkt wird. Das zurückgestrahlte Licht wird mit einer Photodiode gemessen. Die abwechselnd verschieden langen und hohen Helligkeitssignale werden als Striche bzw. Lücken unterschiedlicher Breite interpretiert. Eine andere

Variante ist das Scannen mit einer Diodenleiste (Linienkamera), die über das Etikett geführt wird. Das Bild des Etiketts wird erst im Rechner aus den einzelnen Streifen zusammengesetzt. Der Einsatz einer CCD-Kamera ist eher selten.

Alternativen zu optisch lesbaren Etiketten sind **Transponder** oder **RFIDs** (Radio Frequency Identification). Ein Schaltkreis auf einem Chip (nur Millimeter groß) empfängt Energie von einem Sender und kann damit selbst einen einprogrammierten Code zurückstrahlen, der dann empfangen wird. Die übliche Frequenz ist 13,56 MHz. Die Preise liegen im Centbereich.

Der **EAN-8-Code** besteht aus

- einer 3-stelligen Länder- bzw. Firmennummer,
- einer 4-stelligen Artikelnummer
- einer Prüfziffer

	Zifferncodes			Breite
	Ziffer	odd	even	
Immer zwei Linien und zwei Lücken codieren eine Ziffer. Zwei identische Randzeichen begrenzen links und rechts das Ziffernfeld, ein Trennzeichen steht in der Mitte. Im linken Ziffernfeld beginnt der Zifferncode mit einer Lücke, im rechten Ziffernfeld mit einem Strich. Lücken und Striche treten in den Stärken (Breiten) 1, 2, 3, 4 auf. Die Breiten beziehen sich auf die Breite eines 1-er-Strichs, wie er bei den Randzeichen oder dem Trennzeichen benutzt wird. Die Gesamtbreite eines Zifferncodes ist immer 7.	0	3211	1123	7
	1	2221	1222	7
	2	2122	2212	7
	3	1411	1141	7
	4	1132	2311	7
	5	1231	1321	7
	6	1114	4111	7
	7	1312	2131	7
	8	1213	3121	7
	9	3112	2113	7

Geht der Laserstrahl von links nach rechts über das Etikett, dann gelten die odd-Codes, geht er von rechts nach links, dann gelten die even-Codes. Man kann demnach ein Etikett in beide Richtungen scannen, aber nicht vertikal.

**Die Prüfziffer** wird folgendermaßen gebildet: Beginnend bei der Ziffer vor der Prüfziffer wird rückwärts ( von rechts nach links ) jede Ziffer mit einem Gewicht multipliziert und diese 7 Produkte dann zu einer Prüfsumme addiert. Die 7 Gewichte sind 3, 1, 3, 1, 3, 1, 3. Die Prüfziffer ist dann die Ergänzung der Prüfsumme zur nächsten Zehn. Beispiel: Die Zahl 2007817 gibt die Prüfsumme

$$7 \cdot 3 + 1 \cdot 1 + 8 \cdot 3 + 7 \cdot 1 + 0 \cdot 3 + 0 \cdot 1 + 2 \cdot 3 \Rightarrow 59.$$

Die Prüfziffer ist dann die Ergänzung zur nächsten Zehn, hier also  $60 - 59 = 1$ .

<b>Beispiel Milchpackung</b> von Lidl im EAN-8-Code. Besteht aus einem Startzeichen (111), gefolgt von 4 Ziffern, einem Trennzeichen (1111), gefolgt von weiteren drei Ziffern und der Prüfziffer, abgeschlossen durch ein weiteres Startzeichen (111). Die linken Zeichen beginnen mit einer Lücke, die rechten mit einem Strich. 200 ist die Codenummer für Lidl, 7817 die Milchpackung, 1 die Prüfziffer.	
--	--

(Leider stimmen in meiner Graphik die Linienbreiten und Lückenbreiten nicht 100%ig, sorry)

**Literatur:** Oliver Rosenbaum, Das Barcode-Lexikon  
Bernhard Lenk, Handbuch der automatischen Identifikation

## 6.2 Bildverarbeitungssysteme

### 6.2.1 Anwendungsbeispiele:

- Thermographie im Bauwesen (Wärmebrücken erkennen), heiße Stellen im Leiterplattenbau erkennen
- Qualitätskontrolle: Saubere, intakte Flaschen, intakte Tablettenform, Vollständigkeit einer Leiterplatte
- Computertomographie in der Medizin, Zellcounter bei Blutbildern
- Zeichenerkennung beim Postversand
- Erkennung von Verkehrssituationen beim Fahrassistenten

### 6.2.2 Grundlagen

#### Farbsysteme

Bei selbst leuchtenden Farben (Bildschirm) nimmt man additive Systeme, z.B. <b>RGB</b> (Rot, Grün, Blau)	Bei Auflicht (Druckerzeugnisse) nimmt man subtraktive Systeme, z.B. <b>CMYK</b> (Cyan, Magenta, Yellow, Key (Schwarz))
Grün + Blau → Cyan (Blaugrün) Blau + Rot → Magenta (Bordeaux) Rot + Grün → Gelb Rot + Grün + Blau → Weiß	Cyan + Magenta reflektieren Blau Magenta + Yellow reflektieren Rot Yellow + Cyan reflektieren Grün Cyan + Magenta + Yellow refl. Dunkelgrau

YUV - Farbmodell zerlegt die Farbe in Luminanz (Y= Lichtstärke pro Fläche) und zwei Chrominanz U und V nach den Formeln

$$Y = 0,299 R + 0,587 G + 0,114 B, \quad U = (B-Y) 0,493, \quad V = (R-Y) 0,877 .$$

**Die Attribute** eines Bildes sind:

Dimension (z.B. 1024 x 1024 Pixel)

Grauwert oder Farbe, Farbtiefe (8, 16, 24, 32 Bit, d.h. die Anzahl der Farben und Sättigungen)

Palette, z.B. RGB oder CMYK

Rand oder ohne Rand (Rand oft für Bildoperationen nötig)

Auflösung, meist in Pixel pro Zoll (Dots per inch)

Dateiformat, z.B. Bitmap, JPEG, GIF, TIF

#### Kameratypen:

CCD-Linienkamera enthält nur eine CCD-Leiste. Das Bild entsteht aus der Bewegung.

Scanner arbeiten z.B. nach dem Prinzip der Linienkamera.

CCD-Array-Kamera enthält ein Matrixförmiges CCD-Feld mit Photosensoren. Für Farbaufnahmen ergeben drei Bildpunkte erst die Farbe. Ein *Bayer-Pattern* filtert vor jedem Bildpunkte die richtige Farbe heraus. Ein Farbbild entsteht demnach aus 3 Einzelbildern.

**CCD (Charge Coupled Device):** 1969 von G. Smith und W. Boyle erfunden). Photonen lösen Elektronen aus einer kleinen Halbleiterfläche. Die Elektronen wandern zum nächstgelegenen Kondensator, wo sie gesammelt werden. Von Zeit zu Zeit werden die Kondensatoren ausgelesen (Grauwertsignal der Helligkeit auf einem Pixel). Die Größe eines Pixels (Bildpunkt) ist bei CCD-Kameras vergleichbar der Lichtwellenlänge (1  $\mu\text{m}$ ).

**Matrix:** Bilder werden im Rechner wie Matrizen behandelt. Ein Matrixelement ist die Helligkeit einer Farbe zu diesem Bildpunkt. Meist nimmt man für eine Farbe 256 Grauwertstufen (Helligkeiten) an. Das ergibt 8 Bit = 1 Byte pro Pixel und Farbe. Ein 1024 x 1024 Farbbild würde demnach 3 MByte Speicher benötigen. Durch spätere Bildbehandlung kann jedoch auf Kosten der Farbtreue Information eingespart werden. Fast alle Dateiformate machen das.

Die 8 Nachbarn eines Pixels werden in der mathematisch positiven Umlaufrichtung von 0 bis 7 nummeriert. Pixel 0, 2, 4, 6 heißen direkte Nachbarn, die Pixel 1, 3, 5, 7 heißen indirekte Nachbarn.	3	2	1
	4	<b>P</b>	0
	5	6	7

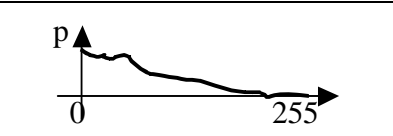
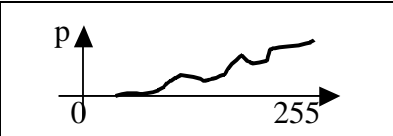
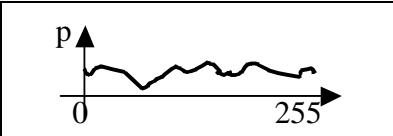
### 6.2.3 Globale Charakteristika eines Bildes

**Mittlerer Grauwert:** Summe aller Grauwerte geteilt durch die Pixelzahl gibt Mittelwert  $m_I$ . Das I steht hier für Imaging (Bildverarbeitung).

**Varianz:** Gibt einen Hinweis auf Helligkeitsunterschiede.  $q_I = \frac{1}{M \cdot N} \sum_m \sum_n (I_{mn} - m_I)^2$

M= Zeilenzahl, N= Spaltenzahl,  $I_{mn}$  = Grauwert bzw. Helligkeit eines Pixels

**Das Histogramm** listet über den 256 Grauwertstufen von 0 (Schwarz) bis 255 (Weiß) die Häufigkeit der Pixel mit genau diesem Grauwert auf. Funktion  $p(g)$  zeigt die relative Häufigkeit  $p$  der Pixel mit Grauwert  $g$ .

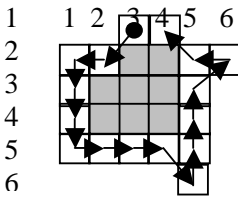
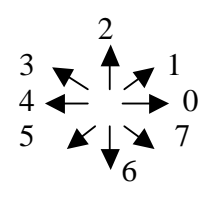
		
Zu dunkles Bild mit wenig Kontrast	Zu helles Bild mit wenig Kontrast	Gut verteilte Helligkeiten und Dunkelheiten - Kontrast

**Die Entropie**  $H = - \sum_g p(g) \log_2(p(g))$  berechnet sich als negative Summe über alle

256 Graustufen  $g$ , wobei die Produkte  $p(g) \cdot \log_2(p(g))$  summiert werden. Der Logarithmus dualis (Logarithmus zur Basis 2) lässt sich auch als  $\ln(p(g)) / \ln(2)$  berechnen. Die Entropie ist insofern interessant, als  $|H|$  die Mindestzahl von Bits liefert, mit der ein Pixel codiert werden kann. Ein Schwarz-Weiß-Bild hat  $H=1$ , ein Bild mit einer sehr guten Grauwertverteilung (siehe oben rechts) hat etwa  $H=8$ .

### 6.2.4 Bildkompression (Beispiele)

**Laufängencodes:** Man zählt die Zahl gleichheller Pixel, die in einer Zeile aufeinander folgen, und speichert nur diese Zahlen  $n$  und die zugehörigen Helligkeiten  $g$  ab. Besonders bei Schwarz-Weiß-Bildern (z.B. gescannter Text oder Zeichnungen) ist die Einsparung erheblich.

<p>Richtungscodes: Man codiert, ausgehend von einem Startpixel, hier [1, 3], mittels der Richtungscodes aus der Graphik ganz rechts die Kontur. Der vollständige Richtungscode der Umrandung ist [1, 3], 5, 4, 6, 6, 6, 0, 0, 0, 7, 2, 2, 2, 1, 4, 3.</p>		
---	--	---

Mit dem Richtungscode kann man:

- Die Figur verschieben, indem man den Startpunkt ändert
- Um jeweils  $45^\circ$  drehen, wenn man zu den Codes 1 addiert modulo 8
- Höhe und Breite einer Figur ermitteln (Minima und Maxima der Zeilen- / Spaltennummern durch Addition ausrechnen)
- Klassifikation einer Kontur in konvex / nichtkonvex

### 6.2.5 Bilddateien (Dateiformate)

Vektorformate werden fast nur bei Konstruktionszeichnungen verwendet: Anfang und Richtung von Linien, Eckpunkte von Rechtecken, Mittelpunkt und Radius von Kreisen, Strichstärken, Strichart, Farbe usw.

Metafile-Formate: Beschreibungssprachen für spezielle Bilder, z.B. Schaltkreise, Raumgitter von FEM-Methoden (Finite Elemente Methode).

Palette: Meist ein Vektor mit 256 definierten Farben (zu je 3 oder 4 Byte codiert). Die Pixelinformation im Bildpunkt bezieht sich auf eine der Palettenfarben. Hier reicht 1 Byte pro Pixel.

BMP (Bitmap von Microsoft): Header ‚BM‘. Palette benötigt 4 Byte pro Farbe (R, G, B, free). Pixelinformation startet mit dem Pixel links unten, dann zeilenweise aufwärts. Schwarz-Weiß (1 Bit pro Pixel), 16 Farben (1 Byte für 2 Pixel), 256 Farben (1 Byte pro Pixel), RGB (3 Byte pro Pixel).

TIF (Tag Image File): Die Header-Information hat keine festen Längen und damit keine festen Positionen, sondern besteht aus einzelnen Informationshappen (Tags), die eine Codenummer haben und jeweils selbst ihre eigene Länge angeben.

JPEG (Joint Photographic Expert Group) erlaubt Kompressionsraten bis zu 90%. Zerlegt die RGB-Codierung der Farben in Luminanz (Y) und die beiden Chrominanz U und V als Koordinaten im YUV-Farbmodell. Jeweils ein  $8 \times 8$ -Pixelfeld wird für sich codiert und gespeichert. Die beiden Chrominanz werden nur als jeweils  $4 \times 4$ -Matrix gespeichert. Die Luminanz wird mit

einer Kosinustransformation in 64 Kosinuskoeffizienten verwandelt, diese gerundet, die kleinen Koeffizienten Null gesetzt, dann mit Lauflängencodes codiert, da viele Nullen auftreten.

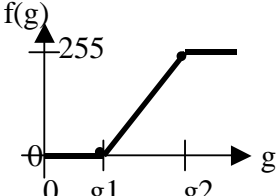
### 6.3 Punktoperationen

Wann benutzt? Bei zu wenig Kontrast, zu dunkel oder zu hell, seitlicher Verschattung

#### 6.3.1 Homogene Punktoperationen

**Lookup-Table (LUT):** Vektor mit 256 Elementen. Jedes Element gibt den neuen Grauwert an, der verwendet werden soll. Beispiel: auf Element 117 steht eine 132. Das bedeutet, dass Pixel mit Grauwert 117 den neuen Grauwert 132 erhalten (werden in diesem Beispiel etwas heller).

**Lineare Grauwertskalierung.** Man verwendet eine Geradengleichung  $f(g) = (g + c_1) \cdot c_2$ . Dabei ist  $f(g)$  der neue Grauwert,  $g$  der alte Grauwert,  $c_1$  eine konstante Grauwertverschiebung,  $c_2$  ein konstanter Kontrastfaktor.  $c_1 > 0$  macht heller,  $c_2 > 0$  verstärkt Kontraste. Werte  $f(g) > 255$  werden auf 255 (weiß) gesetzt, Werte von  $f(g) < 0$  werden auf 0 gesetzt (schwarz).

<p><b>Spreizung eines Grauwertintervalls</b> zwischen den Grauwerten <math>g_1</math> und <math>g_2</math> auf die maximal mögliche Spanne von 0 bis 255 durch eine Geradengleichung mit den Konstanten <math>g_1</math> und <math>g_2</math>.</p>	$f(g) = \begin{cases} 0, & \text{wenn } g < g_1 \\ \frac{g - g_1}{g_2 - g_1} \cdot 255 & \text{sonst} \\ 255, & \text{wenn } g > g_2 \end{cases}$	
--	---	--

**Weitere lineare Skalierungen** können gezielt statt eines alten Grauwertmittels  $m$  und Varianz  $q$  ein neues Grauwertmittel  $m'$  und eine neue Grauwertvarianz  $q'$  einstellen. Man benutzt z.B. die lineare Funktion  $I'_{mn} = (I_{mn} + c_1) \cdot c_2$  mit den Pixelgrauwerten (Intensitäten)  $I_{mn}$  und den Konstanten  $c_1$  und  $c_2$ .

$$m' = \frac{1}{MN} \sum_m \sum_n (I_{mn} + c_1) c_2 = \frac{1}{MN} [(\sum_m \sum_n I_{mn}) c_2] + c_1 c_2$$

oder

$$m' = m c_2 + c_1 c_2. \quad (\text{Gl. 1})$$

Die neue Varianz ist

$$q' = \frac{1}{MN} \sum_m \sum_n ((I_{mn} + c_1) c_2 - (m c_2 + c_1 c_2))^2$$

oder

$$q' = \frac{1}{MN} \sum_m \sum_n (I_{mn} c_2 + c_1 c_2 - m c_2 - c_1 c_2)^2$$

oder

$$q' = \frac{1}{MN} \sum_m \sum_n ((I_{mn} - m) c_2)^2 = \frac{1}{MN} \sum_m \sum_n (I_{mn} - m)^2 c_2^2 = q \cdot c_2^2. \quad (\text{Gl. 2})$$

Wir haben demnach zwei Gleichungen für zwei Unbekannte  $c_1$  und  $c_2$ . Aus Gl. 2 folgt direkt  $c_2 = \sqrt{q'/q}$  und aus Gl. 1 folgt  $c_1 = m'/c_2 - m$ .

**Nichtlineare Grauwertskalierung:** Weit verbreitet ist die **Gammakorrektur**

$$LUT(g) = \left( \frac{g}{g_{\max}} \right)^{1/\gamma} \cdot g_{\max}$$

mit  $g_{\max}=255$ . Im Fall  $g=0$  wird  $LUT(0)=0$  gesetzt. Die  $\gamma$ -Werte bewegen sich in der Praxis zwischen 1,5 und 2,5. (LUT = Lookup Table)

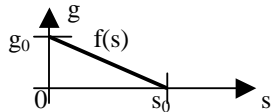
Eine etwas umstrittene Methode ist das **Histogram-Balancing**. Hier werden unterrepräsentierte Grauwertstufen auf Kosten von benachbarten Grauwerten, die überrepräsentiert sind (d.h. zu häufig auftreten) vermehrt, d.h. es werden Pixel umgefärbt.

### 6.3.3 Inhomogene Punktoperationen

Diese Operationen hängen von der Pixelposition ab. Man benutzt sie bei

- Ungleicher Ausleuchtung der Szene, z.B. links zu dunkel
- Schatten in den Ecken
- Ungleichmäßige Sensibilität der CCD-Elemente (Korrektur von Herstellungstoleranzen).

Im Falle von einseitiger Beschattung nimmt man eine lineare Korrekturfunktion  $f(s)$ , z.B.

<p><math>s</math> ist z.B. Spaltennummer des Pixels, (<math>s = 0, 1, 2, \dots, 1023</math>). Konstante <math>g_0</math> ist die maximale Grauwertanhebung am linken Rand. Sie endet bei Spalte <math>s_0</math>.</p>	$f(s) = \begin{cases} 0, & \text{wenns } s > s_0 \\ g_0 - \frac{g_0}{s_0} s & \text{sonst} \end{cases}$	
---	---	---

## 6.4 Fouriertransformation

Dieses aufwendige mathematische Instrument verwendet man

- Bei periodischen vertikalen oder horizontalen Störungen im Bild
- Zur Erkennung von Aliasing-Störungen.

Das Bild wird mittels der diskreten Fourieranalyse in eine Matrix von Fourierkoeffizienten umgerechnet. Dort nimmt man Korrekturen vor, z.B. Löschen der Koeffizienten mit hoher Nummer wirkt wie ein Tiefpass (Weichzeichner), Löschung der Koeffizienten mit kleiner Nummer wie ein Hochpass.

Aliasing-Störungen entstehen, wenn ein gepixeltes Bild, z.B. ein gerastertes Zeitungsbild, mit einer CCD-Kamera aufgenommen wird. Mal trifft ein Zeitungspunkt auf ein Kamerapixel, mal nicht. Es entstehen Ausschlussmuster, die das Bild verfälschen.

**Weichzeichner:** Man faltet das Bild  $g(x)$  z.B. mit einer Gausskurve  $h(x)$ . Die Faltung (Convolution) ist das Integral

$$g(x) \otimes h(x) = \int_{t=-\infty}^{+\infty} g(t) \cdot h(x-t) dt$$

Der Vorgang ist nicht oder nur schwer umkehrbar. Ein unscharf aufgenommenes Bild lässt sich nur bedingt verbessern. Ausnahme, man hat ein Video mit sehr vielen Einzelbildern. Hier kann man die gesuchte Informationen aus vielen Bildern zusammensuchen und mitteln.

## 6.5 Filter

Filter verwenden auch die Umgebung eines Pixels. Eine Maske, bei der das betrachtete Pixel im Mittelpunkt liegt, dient als Matrix von Gewichten. Randpixel müssen gesondert behandelt werden.

### 6.5.1 Lineare Filter

Filter sind eine Funktionen der Grauwerte unter der Maske. Beispiel ist ein Pixel mit Grauwert 24, umgeben von 8 weiteren Pixeln mit unterschiedlichen Grauwerten. Der neue Pixelgrauwert, der in einer Kopie des Bildes statt der 24 eingesetzt wird, ist 21.

$$\begin{array}{l} \text{Grau -} \\ \text{werte} \end{array} \begin{array}{ccc} 17 & 21 & 20 \\ 19 & 24 & 22 \\ 20 & 19 & 18 \end{array} \text{ mal Maske } \frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \text{ gibt } \begin{array}{l} (1 \cdot 17 + 2 \cdot 21 + 1 \cdot 20 + \\ 2 \cdot 19 + 4 \cdot 24 + 2 \cdot 22 + \\ 1 \cdot 20 + 2 \cdot 19 + 1 \cdot 18) / 16 \\ = 21 \end{array}$$

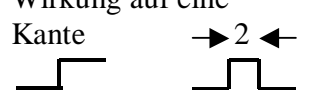
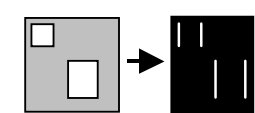
### 6.5.2 Tiefpassfilter

Diese Filter wirken wie Weichzeichner. Homogene Flächen bleiben unverändert.

Der Rechteckfilter macht aus einer Kante eine gestufte Rampe, aus einer 1-Pixelstörung (Fliegendereck) einen verwaschenen Punkt	Der Gaußfilter ist die Approximation einer 2-dimensionalen Gaußkurve durch ganzzahlige Gewichte (Pascalsches Dreieck)
$\frac{1}{9} \begin{array}{ c c c } \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$	<p>Pascal</p> $\begin{array}{c} 1 \\ 1 \ 1 \\ 1 \ 2 \ 1 \end{array} \quad \begin{array}{ccc} & 1 & 2 & 3 \\ 1 & \boxed{1} & \boxed{2} & \boxed{1} \\ 2 & \boxed{2} & \boxed{4} & \boxed{2} \\ 3 & \boxed{1} & \boxed{2} & \boxed{1} \end{array} \frac{1}{16}$

### 6.5.3 Hochpassfilter

Diese Filter verstärken Details, machen Kanten schärfer, machen homogene Flächen schwarz (Grauwert 0). Sie sind wichtig für Mustererkennung oder Teileerkennung.

<b>X-Gradientenfilter</b> (Prewittfilter). Eine Kante wird zu einer 2 Pixel starken Linie	$h_x = \begin{array}{ c c c } \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$	Wirkung auf eine Kante 	
--	--	--	---

Dieser Filter arbeitet nur in x-Richtung gut, d.h. nur senkrechte Kanten werden zu Linien. Die Erweiterung auf horizontale oder schräge Kanten gelingt mit den **Sobelfiltern**

Arbeitet in x-Richtung am besten	Arbeitet in y-Richtung am besten	Arbeitet diagonal links unten nach rechts oben	Diagonal links oben nach rechts unten
$S_x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}$	$S_y = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$	$S_d = \begin{vmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{vmatrix}$	$S_d = \begin{vmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{vmatrix}$

**Laplacefilter** sind sensibel gegen Pixelstörungen (Fliegendreck), aber sie arbeiten in jede der 8 Richtungen gut.

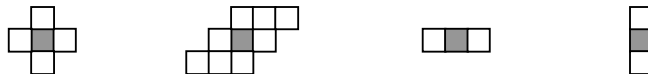
$\begin{vmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{vmatrix}$	oder	$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$	$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix}$
Homogene Flächen werden schwarz (Grauwert 0). Kanten werden verstärkt (heller)		Homogene Flächen bleiben unverändert. Kanten werden verstärkt (heller)	

### 6.5.3 Nichtlineare Filter

Als Beispiel sollen hier **Rangordnungsfiler** dienen. Die Pixelintensitäten (Grauwerte unter der Maske) werden nach der Größe sortiert. Der Grauwert des zentralen Pixels wird in der Kopie des Bildes ersetzt durch das Maximum ( das Minimum, den Median (mittelsten Wert)) der geordneten Grauwerte. Isolierte Defekte werden entfernt. Kanten und lineare Grauwertverläufe bleiben unverändert.

### 6.5.4 Morphologische Operationen

Diese Filter verändern die Konturen von abgebildeten Figuren. Meist werden sie nur auf Schwarz-Weiß-Bilder angewendet. Man benutzt etwa folgende Masken (das zentrale Pixel ist grau):



Dilation (Ausweitung): Wählt die maximale Graustufe unter der Maske. In der Kopie sind die hellen Bezirke vergrößert.	Erosion (Aushöhlung): Wählt die minimale Graustufe unter der Maske. In der Kopie sind die dunklen Bezirke vergrößert.	Closing: Erst eine Dilation, dann eine Erosion. Kleine schwarze Punkte in weißen Flächen verschwinden.	Opening: Erst eine Erosion, dann eine Dilation. Kleine weiße Objekte verschwinden.

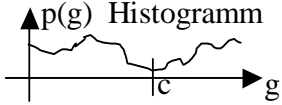
Die Konturen in der Kopie sind nach diesen Operationen nicht mehr so scharf, sondern verwaschen und begradigt.

## 6.6 Bildsegmentation

Die Segmentation ist eine Vorstufe zur Klassifikation bzw. Mustererkennung. Sie eliminiert den Hintergrund und erleichtert das Zählen und Ausmessen von Objekten.

### 6.6.1 Elementare Segmentation

**Thresholding:** Umwandlung eines Grauwertbildes in ein Schwarz-Weiß-Bild mittels eines Schwellenwertes  $c$  (Threshold). Ist  $g$  ein Grauwert z.B. aus  $0...255$ , dann ist  $g_{neu}$  der Schwarz-Weiß-Wert 0 bzw. 1.

<p><b>Den Schwellwert <math>c</math></b> findet man am besten im Histogramm etwa in der Mitte an einer Stelle mit wenig Pixeln zu diesem Grauwert <math>g</math>.</p>	$g_{neu} = \begin{cases} 0, & \text{wenn } g \leq c \\ 1 & \text{sonst} \end{cases}$	 <p>Das Diagramm zeigt ein Histogramm mit der vertikalen Achse <math>p(g)</math> und der horizontalen Achse <math>g</math>. Die Kurve stellt die Verteilung der Grauwerte dar. Ein vertikaler Strich markiert den Schwellwert <math>c</math> auf der horizontalen Achse.</p>
---	--	---

**Lokale Anpassung des Schwellenwerts:** Ist ein Schwellenwert  $c$  nicht optimal, dann passt man den Schwellenwert an die unmittelbare Umgebung des betrachteten Pixels an, z.B. mit der Formel  $c = c_0 - w \cdot m$ . Hierbei ist  $c_0$  der globale Schwellenwert (siehe oben),  $m$  ist die mittlere Intensität (Grauwertmittel) der Nachbarpixel,  $w$  ist ein zu wählendes Gewicht. Der Effekt bewirkt, dass innerhalb heller Objekte die Schwelle sinkt. Es entstehen weniger schwarze Löcher. Außerhalb heller Objekte steigt die Schwelle. Es entstehen weniger weiße Pixel.

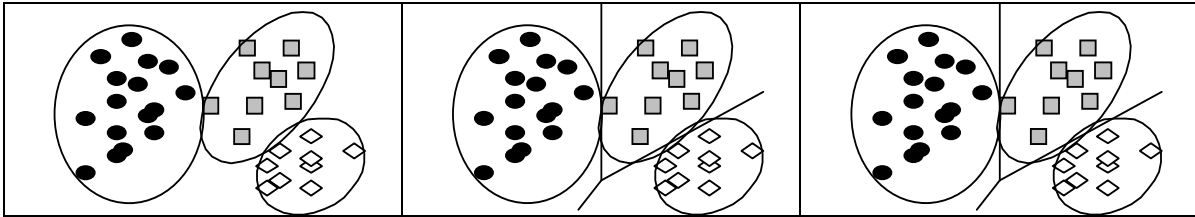
### 6.6.2 Segmentation durch Klassifikation

Wird benutzt bei Zeichenerkennung, Qualitätskontrolle, Produktionsfehlererkennung, Zellcountern. Man kann Objekte trennen, wenn man ihre Gestalt kennt. Voraussetzung ist, dass man aus dem Bild möglichst viele Informationen zieht, die unabhängig voneinander sind. Beispiele sind Helligkeit, Farbanteil, Helligkeitsabnahmen, -zunahmen (Gradienten), lokale Kontraste, topologische Position (z.B. innerhalb einer hellen Fläche). Diese Information zu den Pixeln wird in Variablen gespeichert. Jedes Pixel hat auf diese Weise mehrere Variablenwerte.

Beispiel: Pixelklasse 1 (hell und rot), Pixelklasse 2 (dunkel und grünblau), Pixelklasse 3 (hell und grünblau). Später könnte sich etwa herausstellen, dass Klasse 1 hauptsächlich Pixel von Bakterienabbildungen sind, Klasse 2 Hintergrund, Klasse 3 Pixel von Sporenabbildungen.

Es gibt drei Hauptverfahren der eigentlichen Klassifikation:

<p>Die Clusteranalyse sucht die Klassen, indem sie Häufungspunkte willkürlich als Klassen identifiziert</p>	<p>Die Diskriminanzanalyse bildet mit Hilfe von bekannten Lernobjekten Trennlinien, die die Klassen trennen</p>	<p>Neuronale Netze vermögen dasselbe zu leisten, wie die Diskriminanzanalyse. Aber auch sie müssen lernen.</p>
---	---	--



In den drei Graphiken könnte z.B. die x-Achse die Helligkeit, die y-Achse der Rotanteil in der Pixelfarbe sein.

## 6.7 Kanten und Linien

Sie dienen dem Ausmessen von Durchmessern, Längen usw., der Identifikation von Objekten unabhängig von der Lage und Ausrichtung, der Vektorisierung von gescannten Konstruktionszeichnungen, der Speicherung von Fingerabdrücken usw.



### 6.7.1 Skelettierungsfunktion

Diese Funktion reduziert ein Objekt von einer Fläche auf ein Liniengebilde. Skelettierungsmethoden benutzen fast immer Schwarz-Weiß-Bilder. Die Algorithmen benötigen einen freien Rand von 1 oder 2 Pixel Breite um das Bild.

Es gibt die drei Pixelklassen **0** (Hintergrund), **1** (Vordergrund), **L** (ausgewählt zum Löschen).

Die Pixel genau über oder genau neben einem Pixel heißen direkte **Nachbarn** (d).

Die Pixel schräg darüber oder schräg darunter heißen indirekte Nachbarn (i).

Ein **Pfad** ist eine Folge benachbarter 1/L-Pixel  $P_1, P_2, \dots$

Ein einfacher Pfad enthält ein Pixel nicht mehrmals.

Ein direkter Pfad besteht nur aus direkten Nachbarn (d).

Ein geschlossener Pfad führt zum Startpixel zurück.

i	d	i	Num-	3	2	1
d	P	d	erier-	4	P	0
i	d	i	ung	5	6	7

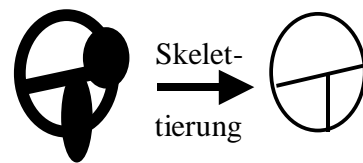
Eine **Verbindung** (Connection) ist eine Pixelmenge, in der jedes Pixel mit jedem anderen durch einen (direkten) Pfad verbunden ist.

Ein **Gebiet** (area) ist eine verbundene Pixelmenge von 1-Pixeln oder ein einzelnes 1-Pixel.

Eine **Kontur** ist eine Pixelmenge, wobei jedes Pixel mindestens ein 0-Pixel als direkten Nachbarn hat. Innere Pixel sind die Pixel, die nicht zur Kontur gehören.

Eine **Linie** ist ein Gebiet ohne innere Pixel.

Ein **Loch** (hole) ist ein Gebiet mit einem geschlossenen Pfad, der eine Menge von 0-Pixeln einschließt.



### 6.7.2 Hough-Transformation

Unterstützt die Suche nach bestimmten Objekten, z.B. Linien oder Kreisen, auch wenn deren Abbilder gestört vorliegen. Hough definiert einen Parameterraum. Ein Objekt wird durch die Häufung von Punkten im Parameterraum definiert.

Beispiel: In der Hesseschen Normalform ist eine Gerade durch ihren Mindestabstand  $r$  vom Ursprung und dem Winkel  $\alpha$  mit der x-Achse definiert. Wir haben demnach 2 Parameter,  $r$  und  $\alpha$ . Man quantisiert den Parameterraum (eine Matrix mit Nullen belegt), indem man nur eine bestimmte Anzahl an Abständen und Winkeln zulässt, z.B. 1-mm-Schritte für  $r$  und  $1^\circ$ -Schritte für

$\alpha$ . Legt man durch jedes Vordergrundpixel (1-Pixel) alle 360 möglichen Geraden und zählt zur entsprechenden Zelle des Parameterraumes ( $r$  und  $\alpha$  bestimmen die Zelle) eine 1 dazu, dann häufen sich auf einer bestimmten Zelle die Eintragungen, falls viele Pixel des originalen Bildes wirklich auf einer Geraden liegen. Die anderen Parameterzellen sind dagegen nur wenig frequentiert. Folglich definiert die hochfrequentierte Parameterzelle eine Gerade mit einem bestimmten Abstand zum Ursprung und einer bestimmten Neigung. Was noch fehlt und in der Praxis nicht einfach zu bestimmen ist, sind Startpunkt und Ende der Geraden.

## AT-Klausur SS 2007

Name:

Doz. Dr. S. v. Weber, FH Furtwangen, FB MuV, 121 Punkte

1. ( 30 P) Programmieren Sie nach dem Beispiel "Autopilot" ein Programm, das Messwerte in einer Schleife liest bis der Schalter "Messung\_aktiv" ausgestellt wird. Messwerte lesen Sie etwa so:

```
x=Eingabe_Messwert ( Kanal1); // Der gelesene Messwert steht dann auf x
```

Vor der Schleife setzen Sie `Summe_x=0;` und `n=0;` und öffnen Sie die Messwertdatei zum Schreiben mit

```
FILE *aus; aus=fopen("X:\\DATA\\Tempdata.prn","w");
```

Glätten Sie die gelesenen Messwerte in der Schleife mit Exponentieller Glättung ( $C=0.95$ ).

Geben Sie die geglätteten Messwerte auf die Datei `aus` zusammen mit der fortlaufenden Nummer  $n=1, 2, \dots$

```
Benutzen Sie dafür fprintf(aus, "\n n=%6i Temp=%6.2f ", n , xglatt );
```

Warten Sie mit `delay(1000);` nach jeder Messwertausgabe 1 Sekunde.

Bilden Sie den Mittelwert (alle geglätteten Messwerte in der Schleife addieren und zählen und nach Beendigung der Schleife dann dividieren. Geben Sie den Mittelwert ebenfalls auf die Datei `aus`.

2. ( 28 P) Entwerfen Sie den Mealy-Graphen mit seinen Prozessschritten für folgenden chemischen Prozess: Warten, Befüllen des Reaktionsgefäßes mit Baumwolle bis der Arbeiter Signalknopf "voll" drückt, Befüllen mit Schwefelsäure bis Pegelmelder P1, Befüllen mit Salpetersäure bis P2, 10 Minuten Reaktionszeit abwarten (Timer T1) bei gleichzeitiger Kühlung über den Mantel, Ablassen der Säure (P0 abfragen), Wasserzulauf bis Pegel P2, 5 Minuten einweichen lassen (Timer T2), Ablassen des Waschwassers (P0 abfragen), Leuchtsignal "Leeren" für den Arbeiter als Start zum Entleeren, zurück ins Warten wenn von diesem Signalknopf "leer" gedrückt wird

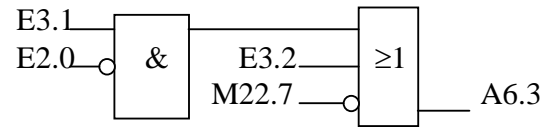
a) Machen Sie eine Anlagenskizze mit Ventilen V1, V2, ..., Kessel, Kühlmantel, Pegelmeldern

b) Zeichnen Sie den Mealy-Graphen mit Überführungs- und Ausgabesignalen

3. ( 18 P) Zeichnen Sie das Symbol eines 5/2-Steuerventils mit Handantrieb (Schlagknopf) und einen doppelt wirkenden Zylinder, der gedrosselt ausgefahren wird und nach dem Loslassen des Schlagknopfes ungedrosselt eingeholt wird (Drossel mit Rückschlagventil in eine der Luftleitungen einzeichnen). Schreiben Sie die Anschlussnummern 1, 2, 3, 4, 5 richtig an das 5/2-Ventil.

4. ( 10 P)

Programmieren Sie nach dem Beispiel der Übung 11 in der Sprache AWL das rechts abgebildete Netzwerk. Zwischenspeicherung des "UND"-Ergebnisses auf Merker M22.0



5. (25 P) Berechnen Sie nach dem Beispiel von Übung 21 in 3 Schritten die Reglermatrix R (Schritt 4 nicht rechnen). Gegeben sind die 4 Sprungantworten auf zwei Stellgrößen sprünge  $\Delta y_1$  und  $\Delta y_2$ :



Schritt 1: Kps-Matrix berechnen und Dimensionen angeben

Schritt 2: Die beiden Haupt-P-Regler  $D_{11}$  und  $D_{22}$  mit  $V_o T_u / T_g = 0.6$  berechnen

Schritt 3: Die inverse Kps-Matrix berechnen und daraus Reglermatrix R der P-Regler mit Dimensionen

6. ( 10 P) Beantworten Sie kurz und knapp folgende 4 Fragen zum Skript

- Was ist eine Schrittschaltwerksteuerung? (unter Steuerungssysteme der Pneumatik)
- Was ist der Unterschied zwischen S- und K-Regler? (unter Datenbaustein für PID-Regelung)
- In welcher Formel taucht Schwellwert  $\theta_j$  beim Einzelschicht-Perzeptron auf? Schreiben Sie sie ab
- Wie lautet die Formel zum Gütemaß  $I^2$  (Quadratische Regelfläche)?

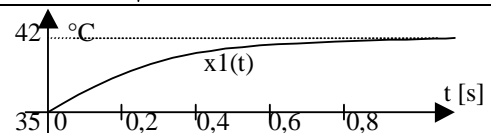
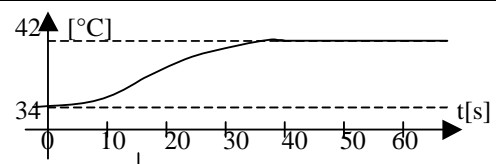
### Klausur Automatisierungstechnik WS 2007/08

Name:

Doz. Dr. S. v. Weber, FB MuV, HS Furtwangen, Gesamt 80 P

1. (15 P) Gegeben ist die Sprungantwort eines Rührkessels auf einen Heizleistungssprung der Höhe  $U_0 = \Delta y = 1,8 \text{ [KW]}$  und die Sprungantwort der Temperaturmessung auf einen Temperatursprung von  $35^\circ\text{C}$  auf  $42^\circ\text{C}$  (Badwechsel des Thermometers)

- Identifizieren Sie die Regelstrecke (Rührkessel) und zeichnen Sie das IAS-Blockschaltbild mit Eintragung der identifizierten Parameter als Zahlen
- Identifizieren Sie die Rückführung (alle Parameter)



2. (10 P) Schreiben Sie die Übertragungsfunktionen der folgenden Bauteile einmal allgemein und dann mit eingesetzten Zahlenwerten auf:

PID-Regler mit  $K_{pr}=19,3$  [KW/K],  $T_N=32$  [s],  $T_V=4,7$  [s]  
 Rührkessel Typ PT3 mit  $K_{ps}=0,77$  [K/KW],  $n=3$ ,  $T=12$  [s]  
 Rückführung Typ PT1 mit  $K_p=1$ ,  $T=2,3$  [s]

**3. (15 P)** Berechnen Sie das Ausgabesignal  $y$  eines Neurons mit den Eingängen  $x_1=2.9$ ,  $x_2=-1.4$ ,  $x_3=4.2$ , den Gewichten  $w_1=0.8$ ,  $w_2=0.9$ ,  $w_3=1.6$ , einem Schwellwert  $\theta=5.0$ , sowie einer Aktivierungsfunktion, die für einen Aktivierungswert  $\alpha < -10$  den Wert  $y = -1$  ausgibt, für  $\alpha > +10$  den Wert  $y = +1$  und dazwischen linear interpoliert, dh. Werte zwischen  $-1$  und  $+1$  ausgibt.

<p><b>4. (10 P)</b> In einer Fuzzy-Regelung soll eine Zuordnungsfunktion mit Namen <math>PHR(pH)</math> in Form einer Auswahlklammer geschrieben werden (siehe ganz rechts). Die Daten entnehmen Sie der Graphik.</p>		$PHR(pH) = \begin{cases} \dots, & \text{wenn} \\ \dots, & \text{wenn} \\ \dots & \text{sonst} \end{cases}$
---	--	--

<p><b>5. (15 P)</b> Schreiben Sie das AWL-Programm zu folgenden beiden Aufgabenstellungen. In der Aufgabe a) nehmen Sie als Zwischenmerker M88.0 als Ausgang der UND-Box.</p>	<p>a) Stelle auf M120.0 einen Einsmerker her. Realisiere gleich danach das folgende Schaltbild:</p>	<p>b) Schreiben Sie den Aufruf von FB250 (Analogeingabe) für eine Sollwerteingabe, davor Aufruf Datenblock 8, Herstellen eines Nullmerkers. Die Angaben zur Eingabe sind: Baugruppe 192, Kanal 4, Kanaltyp 3, Zielintervall [0;2000], keine Einzelabtastung, Fehlerbits auf Merker M7.1 bis M7.3</p>
---	---	--

**6. (15 P)** Der eingestellte Messbereich in einem Messumformer sei  $20 - 80$  °C. Im Fermenter sei augenblicklich eine Temperatur von  $T=36,25$  °C.

a) Berechnen Sie das Messsignal in % und in mA für einen (4 - 20)- mA-Ausgang.

b) Wir nehmen einen 12-Bit-Wandler an mit 4096 möglichen Stufen (0 bis 4095). Bilden Sie das Messsignal auf die Stufen ab (0 % entspricht Stufe 0, 100 % entsprechen Stufe 4095). Runden Sie die Stufennummer zur nächstgelegenen ganzen Zahl.

c) Berechnen Sie den Dualwert der gerundeten Stufennummer. (Falls Sie Aufgabe b) auslassen möchten, dann von der beliebigen Stufennummer 972). Geben Sie das Ergebnis auch hexadezimal an.

## Liste der Übungen aus der Vorlesung

1. Programmierung einer kleinen Prozesssteuerung in C
2. Planung einer kleinen Versuchsanlage (Skizze, Mealygraph)
3. Messgrößen und ihre Sensoren
4. Umrechnung von Einheitssignalen, binäre Zahlen
5. Netztopologien der Kommunikation
6. Schätzung der Energiekosten einer Pneumatikanwendung

7. Logischer Schaltplan aus UND- und ODER-Boxen
8. Abänderung eines gegebenen kleinen Pneumatikplans
9. Kraftentfaltung beim doppelt wirkenden Pneumatikzylinder
10. Wägeverfahren der Analog-Digital-Wandlung
11. Kleine Aufgaben der AWL-Programmierung einer SPS
12. Notation der PID-Reglerparameter als Festkommakonstanten
13. Analogwerteingabe in der SPS mit Funktionsbaustein FB250
14. Umwandlung einer Differentialgleichung in eine Differenzgleichung
15. Rund um den S-Regler (Komponenten, PID-Formel, Begrenzung)
16. Funktionsdefinition einer Zuordnungsfunktion in der Fuzzy-Control
17. Programmierung einer Regel der Fuzzy-Control mit der Min-Max-Methode
18. Exponentielle Onlineglättung gestörter Daten
19. Logische Bedingungen bei der multivariaten Klassifikation
20. Koppelmaß für Regelkreise berechnen
21. Parametrierung von Entkopplungsreglern nach Boksenbom und Hood
22. Parametrierung von einfachen Übertragungsfunktionen
23. Gütemaße der Regelung (Tan, Taus, Xmax, Dämpfung)
24. Blockschaltbild einer Regelung: Identifikation, Programmierung

Übungen der Bildverarbeitung nur bei 3-stündiger Vorlesung

25. Speicherbedarf eines Bildes
26. Spreizen eines Grauwertintervalls
27. Gaußfilter
- 28.

## Wörter der Automatisierungstechnik - Dictionary of Automation

Zur Vorlesung Automatisierungstechnik

Abbruch	abort, program stop	Messumformer	measuring transformer
abfahren (Prozess)	shut-down	Nachstellzeit (Regler)	reset time
Ablaufsteuerung	sequencing control	Netzwerk	ladder contact network
Abtastzeit	sampling time	Nocke (Nockenscheibe)	cam
Aktor	actuator	Pneumatikplan	Pneumatic circuit diagram
Analogwertverarbeitung	analog value processing	pneumatische Steuerung	pneumatic control
anfahren (Prozess)	start, start-up	Programmiergerät	programmer
Anpassung (SPS)adapter	block	Prozessabbild	process I/O image
Anschluss (Schlauch)	connection	Pumpe	pump
Anweisung (Programm)	instruction, statement	Rauchmelder	smoke detector
Anweisungsliste	statement list	Regelgröße	controlled variable
Aufbau	construction	Regelkreis	closed (control) loop
Ausfall	failure	Regelstrecke	controlled member
Ausschaltverzögerung	release delay	Reglerverstärkung	gain
Automatikbetriebautomatic	mode	Relais	electrical relay
Automatisierungseinrichtung	automation system	Richtimpuls	initializing pulse
Automatisier.-funktion	automation function	Rückschlagventil	non-return valve
Automatisierungsgerät	programmable controller	Schalldämpfer	silencer, muffler
Baustein (Programm)	module, block	Schaltbild	connection diagram
Befehl	instruction,command	Schalter	switch
Befehlskette	sequence	Schaltstellung	position
		Schaltwarte	control room, - centre

Beharrungswert	steady-state value	Schaltzeichen	switching symbol
Bereichsüberschreitung	limit error, overflow	Schlauch(kupplung)	tube (coupler)
Betriebsartenteil	operating mode module	Schütz (Relais)	contactor
Blockschaltbild	block diagram	SPC	(storage programmable control).
Dämpfung	damping	SPS	SPC stor. progr.control
Drossel(ventil)	throttle (valve)	Speicherverhalten	storage properties
Druckregler	pressure controller	Spindel (Ventil)	spindle, stem, rod
Düse	nozzle, jet	S-Regler	step controller
Einschaltverzögerung	closing delay	Starttaste	start button, start key
elektropneumatisch	electropneumatic	Staudruck	impact pressure
Endlagenschalter	limit switch	Stellgröße	control variable
Exschutz	explosion protected	Steuerungsaufgabe	control function
feststellen (Spannvorr.)	chuck, clamp, gripp	Steuerwort	control -/command word
Flankenbewertung	pulse edge evaluation	Stoßmagnet	push-type electromagnet
Freigabe Signal, -Luft	release sig., enable air	Taktkette (Ablaufkette)	sequence cascade
Führungsregelung	pilot control, setpoint c.	Totzeit	delay time
FUP Funktionsplan	function Diagram	Turbulenzverstärker	turbulence intensifier
Getriebe	gear, transmission	Überführung (Graph)	transition
Grundstellung (-zustand)	initial position (- state)	Überlastsicherung	overload protection
Handbetrieb	manual mode	Ventil	valve
Kante (Graph)	edge	Verriegelung	interlocking
Klemme, Kontakt	clamp, contact	Verstärker	amplifier
Knoten (Graph)	node, state	Vorhaltezeit (Regler)	rate time
Kolben(stange)	piston (rod)	Weiterschaltbedingung	stepping conditions
Kontaktplan	ladder diagram	Wirkungsgrad	operating ratio
K-Regler	continuous controller	Zähler	counter
LED (Leuchtdiode)	light emitting diode	Zeitfunktion	timing function
Leitstand	control room	Zeitglied	timer
Leitsystem	control system	Zeitkonstante	time constant
Leitung (Strom,Luft)	electric line, air pipe	Zeitplansteuerung	time-program control
Lichtschanke	light barrier	Zeitverhalten	time response,specified t.
logische Verknüpfung	logic operation	zuführen	supply, feed
lösen (Spannvorrichtung)	release, loose	Zugmagnet	pull-type electromagnet
Luftaufbereitung	air conditioning	Zustand (Graph)	state (of graph)
Merker	flag, marker	Zylinder	cylinder